

# Entwicklung der Softwareplattform RESUS: Repository Simulation, Uncertainty propagation and Sensitivity Analysis

## **Dissertation**

Zur Erlangung des Doktorgrades  
der Naturwissenschaften

Vorgelegt von

**Xiaoshuo Li M.Sc.**

aus Shanxi, China

genehmigt von der Fakultät für Energie- und Wirtschaftswissenschaften  
der Technischen Universität Clausthal,

Tag der mündlichen Prüfung  
16.07.2015

Vorsitzender der Promotionskommission

Prof. Dr.-Ing. Norbert Meyer

Hauptberichterstatte

Prof. Dr. rer. nat. Klaus-Jürgen Röhlig

Mitberichterstatte

Prof. Dr. rer. nat. Kurt Mengel

Diese Arbeit wurde anteilig im Rahmen des Forschungsvorhabens „Entsorgungsoptionen für radioaktive Reststoffe: Interdisziplinäre Analysen und Entwicklung von Bewertungsgrundlagen“ (ENTRIA) gefördert.



GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung

Förderkennzeichen: 02S9082A

### Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.



## Inhaltsverzeichnis

	Seite
<b>1</b>	<b>Einleitung</b>
1.1	Ziel der Arbeit
1.2	Hintergrund / Relevanz
<b>2</b>	<b>Vorstellung des aktuellen Stands der vorhandenen Software zur Sicherheitsbewertung eines Endlagers</b>
2.1	AMBER
2.2	GoldSim
2.3	ALLIANCES
<b>3</b>	<b>Einführung in die Software-Plattform RESUS</b>
<b>4</b>	<b>Vorstellung der Funktionalitäten von RESUS (Nutzeranleitung)</b>
4.1	Ishigami-Modell in RESUS
4.2	Level-E in RESUS
4.3	Eindimensionales Transportmodell in RESUS
4.3.1	Allgemeine Vorstellung eines eindimensionalen Transportmodells in RESUS
4.3.2	Ein Beispiel für eine eindimensionale Transportmodellierung
4.4	Zweidimensionales Transport-Modell in RESUS
4.4.1	Allgemeine Vorstellung der zweidimensionalen Transport-Modelle
4.4.2	Ein Beispiel für ein zweidimensionales Transportmodell in RESUS
4.5	Beispiel für die Erstellung eines Analysemodells mit der dateibasierenden Methode
4.5.1	Allgemeine Vorstellung der dateibasierten Modellerstellungsmethode
4.5.2	Beispielmodell zur Erläuterung der dateibasierten Methode in RESUS
4.6	Speichern und Wiederöffnen eines Analysemodells in RESUS
4.6.1	Speichern eines Modells
4.6.2	Dateien zum Speichern der Modellinformationen
4.6.3	Wiederöffnung eines Analysemodells in RESUS aus Datei
<b>5</b>	<b>Entwicklungsdokumentation des Programms RESUS</b>
5.1	Details der Klassen- und Objektsstruktur
5.1.1	Die Klasse „Model“
5.1.2	Die Klasse „SubModel“
5.1.3	Die Klasse „Parameter“

5.1.4	Die Klasse „ResultsParameter“	124
5.2	Andere Klassen für den Modellaufbau	126
5.2.1	Die Klasse „ParameterDeterministic“	126
5.2.2	Die Klasse „Geoformation“	127
5.2.3	Die Klasse „GeologicAnomaly“	129
5.3	Anderen Bestandteile in RESUS	130
5.3.1	Zufallsgenerator	131
5.3.2	Ishigami Modell	133
5.3.3	Level-E Modell	133
5.3.4	Das Transportmodell mit „RockFlow“ und „OGS“	135
5.3.5	Die dateibasierte Konstruktionsmethode (file based method)	143
5.3.6	Datenaufbereitung	149
5.3.7	Die grafische Darstellung	152
5.3.8	Speichern und Wiederöffnen eines RESUS-Modells	155
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>161</b>
<b>7</b>	<b>Anhang</b>	<b>164</b>
<b>8</b>	<b>Tabellenverzeichnis</b>	<b>174</b>
<b>9</b>	<b>Abbildungsverzeichnis</b>	<b>175</b>
<b>10</b>	<b>Literaturverzeichnis</b>	<b>182</b>

# **1 Einleitung**

## **1.1 Ziel der Arbeit**

Die vorliegende Arbeit ist ein Beitrag zur Endlagerforschung. Auf diesem Gebiet wird eine langfristig sichere und nachsorgefreie Endlagerung radioaktiver Abfälle untersucht. Um Aussagen über die Sicherheit eines Endlagers treffen zu können, werden Modelle erstellt und für verschiedene Szenarien analysiert. Die hohen Ansprüche an die Endlagersicherheit erfordern sehr komplexe Modelle, die nur mit Hilfe spezieller Software modelliert und berechnet werden können. An dieser Stelle setzt die vorliegende Arbeit ein.

Kernthema dieser Dissertation ist die Einführung, Vorstellung und Erläuterung eines neuen Software-Produkts namens RESUS, das vom Verfasser dieser Dissertation konzipiert und programmiertechnisch umgesetzt wurde. Mit Methoden der computergestützten numerischen Modellierung dient es der Erstellung und den probabilistischen Analysen der Endlagermodelle. RESUS soll in Lehre und Forschung Einzug finden und für Sicherheitsnachweise von Endlagersystemen herangezogen werden. Diese Arbeit wird mittels eines Nutzerhandbuchs und einer Entwicklungsdokumentation dargestellt.

## **1.2 Hintergrund / Relevanz**

Radioaktive Abfälle sind Stoffe im Sinne des §2 Abs. 1 des Atomgesetzes [1], die nach §9a des Atomgesetzes geordnet beseitigt werden müssen [2]. Die meisten radioaktiven Abfälle entstehen durch den Betrieb und den Rückbau von Kernkraftwerken. Kleinere Mengen kommen aus Medizin, Industrie und Forschung. Einige Staaten haben erhebliche Altlasten aus der Entwicklung und Herstellung von Kernwaffen. Das Gesamtaufkommen an derartigen Abfällen wird für Deutschland bis zum Jahr 2060 auf mehr als 300.000 m<sup>3</sup>, bei Einbeziehung der aus der Schachanlage Asse zurückzuziehenden Abfälle und dem abgereicherten Uran aus der Urananreicherung auf mehr als 600.000 m<sup>3</sup> geschätzt. Mehr als 90 Volumenprozent der bis 2060 prognostizierten Abfälle sind schwach oder mittelaktiver Abfall (LILW – Low/Intermediate Level Waste) oder – nach deutscher Nomenklatur – Abfall mit vernachlässigbarer Wärmeentwicklung [3][63].

Unterschiedliche Wege zur Entsorgung radioaktiver Abfälle wurden seit den 1950er Jahren auf der Welt diskutiert und z.T. auch systematisch bewertet. Hierzu gehört z.B. der Transport in den Weltraum (in die Sonne, auf eine Sonnen- oder Erdumlaufbahn oder auf den Mond), die Verklappung auf den oder unter den Meeresgrund, die Verbringung in Subduktionszonen, in Vulkane, in die Antarktis und die direkte Injektion flüssiger Abfälle in tiefe geologische Schichten. Im Jahr 2006 hat Großbritannien die jüngste systematische Betrachtung und Bewertung solcher Optionen abgeschlossen. Wie schon bei früheren Untersuchungen anderer Länder ist man auch hier zum Schluss gekommen, dass die meisten genannten Optionen aus praktischer Sicht entfallen. Risikoabschätzungen oder Umweltbedenken, rechtliche Einschränkungen und Zweifel an der technischen Machbarkeit oder Wirtschaftlichkeit lassen

diese nicht vertretbar erscheinen. In der Konsequenz wird der langfristige Einschluss der Abfälle in Entsorgungseinrichtungen auf oder unter der Oberfläche als einzig sinnvolle Option betrachtet. Je nach Abfallart gehören zu den Einrichtungen auf oder nahe der Erdoberfläche sowohl regelmäßig überwachte und gewartete Lagerhallen bzw. Nasslager als auch oberflächennahe Endlager in Form von Abfallhalden oder Betonkammern (für kurzlebige Abfälle). Unter der Erde sind bergwerksähnliche Endlager in einigen Dutzend bis einigen hundert Metern Tiefe die möglichen Entsorgungsarten. Insbesondere für hochradioaktive Abfälle sind letztere Endlager in tiefen geologischen Schichten die weltweit bevorzugte Lösung [4][7][8]. Deutschland strebt diese Art der Lagerung überdies für alle radioaktiven Stoffe an [5][6]. Unabhängig von ihrer Bauart haben alle Endlager die gleiche Aufgabe. Sie müssen radioaktive Abfälle und die in ihnen enthaltenen radio- und chemotoxischen Stoffe langfristig sicher von Mensch und Umwelt fernhalten.

Um diese Aufgabe auch in der Zukunft zu erfüllen, ordnen sich alle Aspekte der Endlagerung dem zentralen Gedanken einer passiven Sicherheit unter. Dies bedeutet auch einen Verzicht auf zukünftige Wartung durch den Menschen, kurz: Wartungs- und Nachsorgefreiheit. In keinem Szenario darf es zu einem Versagen der Endlagerfunktion kommen. Sowohl die klassische tiefe geologische Endlagerung, als auch die Endlagerung mit Rückholungsmöglichkeit streben daher Systeme an, die bis in die ferne Zukunft unabhängig von äußeren Eingriffen funktionieren. Diese langfristige Autarkie wird durch so genannte Sicherheitsfunktionen unterstützt. Sie werden von dem ganzen Endlager als Systemverbund gewährleistet und sollen in erster Linie den Einschluss der Materialien und ihren sicheren Verbleib am Einlagerungsort sicherstellen. Doch auch wenn ein Einschluss über längere Zeiträume nicht vollständig möglich sein sollte und sich Schadstoffe mobilisiert haben, muss die Stoffbewegung („Migration“) soweit behindert und verzögert werden, dass keine ernsthaften Umweltschäden zu befürchten sind. Die Schadstoffmigration wird mit Hilfe eines sogenannten Multibarrieren-Systems unterbunden, dass sowohl auf natürliche geologische Barrieren, künstlichen technischen Barrieren (Abfallmatrix, Abfallbehälter usw.) und geotechnische Barrieren (Verfüllung, Puffermaterial, Strecken- und Schachtverschluss usw.) setzt.

Eine entscheidende Rolle in Bezug auf die genannten langfristigen Ziele spielt die Standortwahl. Es liegt nahe, dass die Errichtung eines Endlagers in einer geologisch stabilen Region erfolgen muss. Innerhalb dieser muss eine passende geologische Formation zur Aufnahme des Endlagers gefunden werden. Diese nennt man Wirtsgesteinsformation. International werden drei Typen von geologischen Formationen als Wirtsgestein betrachtet: Salinalgestein, Tonformationen und Kristallingesteine. Für unterschiedliche Wirtsgesteine werden jeweils angepasste Endlagerkonzepte verwendet [9][10][11].

Die durch diese Überlegungen erreichte Sicherheit muss für jedes Endlager dargestellt und begründet werden. Dies erfolgt durch einen sogenannten Sicherheitsnachweis (safety case). Aufbauend auf einer umfassenden Sicherheitsanalyse enthält er die Prüfung und Bewertung von Daten, Maßnahmen, Analysen und Argumenten, die als Beleg für die Einhaltung der Sicherheitsanforderungen dienen. Überdies werden darin alle in den Sicherheitsanforderungen genannten Nachweise zusammengeführt. Ein umfassender Sicherheitsnachweis kann mit einem dem jeweiligen Kenntnisstand entsprechenden Tiefgang für verschiedene Schritte eines Endlagerprogramms (z.B. Konzeption, Standortauswahl, Standorterkundung, Errichtung,

Einlagerung, Verschluss) erfolgen. Es wird insbesondere zwischen Sicherheitsnachweisen für die Betriebsphase und Sicherheitsnachweisen für die Nachverschlussphase des Endlagers unterschieden [9][10][11].

Für die vorliegende Arbeit liefert der zuletzt genannte Sicherheitsnachweis der Nachverschlussphase den übergeordneten Zusammenhang. Der Sicherheitsnachweis für die Nachverschlussphase beinhaltet im Wesentlichen die Ergebnisdarstellung einer Langzeitsicherheitsanalyse. Diese belegt, dass das Endlager innerhalb eines geforderten Nachweiszeitraumes oder länger sicher ist. Eine Langzeitsicherheitsanalyse enthält u.a. die folgenden Elemente:

- System- und Prozessbeschreibung (mit FEP-Sammlung und –Processing)
- Definition und Analyse von Sicherheitsfunktionen
- Entwicklung und Auswahl von Szenarien
- Qualitative und/oder quantitative Analyse mit numerischer Modellierung einschließlich Unsicherheits- und Sensitivitätsanalyse

Die im letzten Punkt aufgeführte numerische Modellierung ist eines der wesentlichen Elemente dieses Nachweises und allgemein eines der wichtigsten Hilfsmittel für die Beschreibung, Prognose und Bewertung eines Endlagersystems. Im Folgenden werden zwei für die Langzeitsicherheitsanalyse wichtige Modelltypen der numerischen Modellierung vorgestellt. Sie sind durch ihren Anwendungsbereich charakterisiert. Es handelt sich um zwei Gruppen: integrierte Modelle und Prozessmodelle.

Das Auffahren des Grubengebäudes und die Einlagerung radioaktiver Abfälle stören den vorherigen Zustand der geologischen Formation. In der Folge finden lange andauernde thermische, hydraulische, mechanische und chemische (THMC), z.T. gekoppelte Prozesse statt. Übliche Prozessmodelle beschreiben diese einzelnen, jeweils für Teilsysteme relevanten THMC-Prozesse.

Integrierte Modelle stellen das Endlager hingegen als ein ganzes System dar und ermöglichen die Bewertung seiner Sicherheit anhand von Sicherheitsindikatoren über einen langen Zeitraum. Integrierte Modelle können die behandelten Prozessmodelle mitunter integrieren. THMC-Prozesse und Kopplungen werden dabei aus praktischen Gründen vereinfacht abgebildet. Die vorgenommenen Vereinfachungen werden erläutert und ihre Zulässigkeit belegt [13].

Da manche der einzulagernden radioaktiven Nuklide sehr langlebig sind, muss die Sicherheit eines Endlagers über sehr lange Zeiträume (lt. [12] über 1 Millionen Jahre) betrachtet werden. Die Entwicklungsmöglichkeiten des Endlagersystems über diese Zeitdauer werden mittels verschiedener Szenarien beschrieben und simuliert. Diese bilden die Grundlage insbesondere des integrierten Modells.

Für beide Typen von Modellen zur Durchführung der Langzeitsicherheitsanalyse werden Unsicherheitsanalysen und Sensitivitätsanalysen durchgeführt. Diese finden Verwendung, um Ausmaß und Auswirkungen von Unsicherheiten und Ungewissheiten zu charakterisieren.

Solche Unsicherheiten ergeben sich z.B. durch die Heterogenität geologischer Formationen, die Komplexität der ggf. gekoppelten THMC-Prozesse und die Unmöglichkeit einer vollständigen Messung und Bestimmung aller für die Langzeitsicherheitsanalyse benötigten Parameter.

Um die Modellierungsprozesse für die Langzeitsicherheitsanalyse zu systematisieren und zu automatisieren wurde eine Reihe von Software-Produkten entwickelt. Dazu gehören z.B. das Programm ALLIANCES [14] der französischen Endlagerorganisation ANDRA, sowie die kommerziellen Programme AMBER [15] und GoldSim [16]. Die grundsätzliche Funktionalität und die Grenzen dieser Software-Produkte werden in Kapitel 2 vorgestellt.

Das Institut für Endlagerforschung der TU-Clausthal entwickelt eine Softwareplattform RESUS (Software platform for Repository Simulation, Uncertainty propagation and Sensivity analysis), die sowohl in der Forschung als auch in Lehre anwendbar sein soll.

In dieser Dissertation werden die Grundlagen und Funktionalitäten der ersten Version von RESUS als Nutzeranleitung (Kapitel 4) und als Entwicklungsdokument (Kapitel 5) erfasst.

## 2 Vorstellung des aktuellen Stands der vorhandenen Software zur Sicherheitsbewertung eines Endlagers

Es gibt auf dem Markt verschiedene Software-Produkte, die zur deterministischen und probabilistischen Simulation der Prozesse in einem Endlagersystem verwendet werden können. Darunter sind sowohl kommerzielle Produkte wie z.B. AMBER und GoldSim, als auch von Instituten entwickelte nicht kommerzielle Werkzeuge wie z.B. die französische ALLIANCES-Software. In diesem Kapitel werden die drei genannten Programme, die für die probabilistische Simulation und Analyse in der Sicherheitsanalyse für Endlager genutzt werden, als Stand der Technik vorgestellt.

### 2.1 AMBER

AMBER ist eine kommerzielle Software zur probabilistischen Simulation, die von „Enviros Consulting Limited“ und „Quintessa Limited“ entwickelt wurde. AMBER wird für die Modellierung von Schadstoffmigration in technischen Systemen und im Untergrund genutzt. Die Software ist darüber hinaus auch zur Simulation der Migration von Radionukliden in einem Endlagersystem geeignet. Mit AMBER kann der Nutzer ein integriertes Modell in Modellkomponenten zerlegen und einzelne Komponenten wie ein Transportmedium betrachten [15].

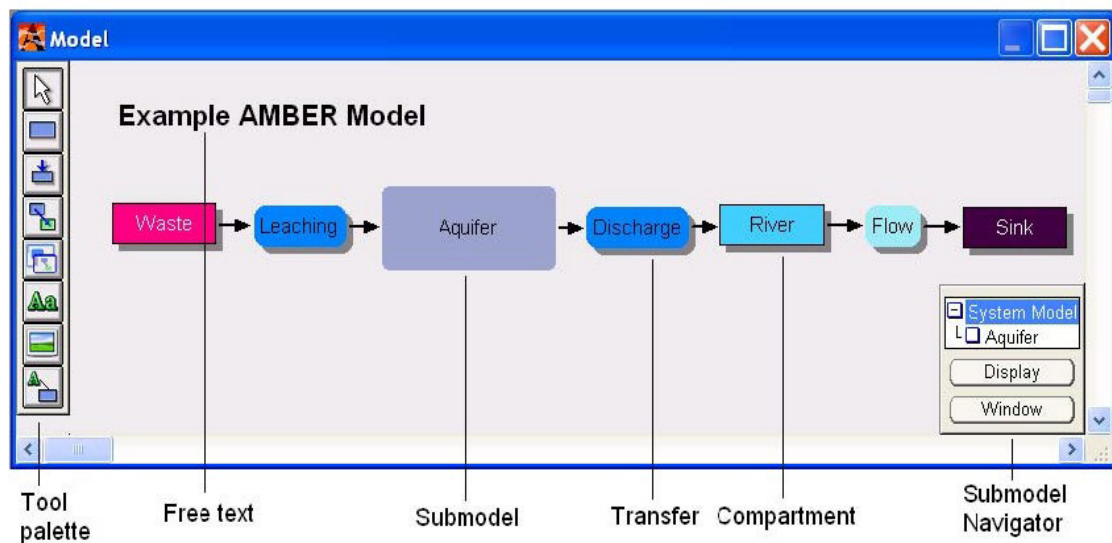


Abb. 2.1 Nutzeroberfläche von AMBER und ein einfaches Beispiel von Schadstofftransport /Bildschirmausschnitt der AMBER-GUI/

Generell wird die molare Stoffkonzentration in jedem Transportmedium mit der folgenden linearen gewöhnlichen Differenzialgleichung berechnet:

$$\frac{dI_i^m}{dt} = -[\lambda_r^m + \sum_j \lambda_{ij}] * I_i^m + \lambda_r^{m+1} I_i^{m+1} + \sum_j \lambda_{ji} I_j^m \quad (2-1)$$

Hier:  $\lambda_{ij}$  [ $y^{-1}$ ] ist die Stofftransportrate zwischen Komponente i und Komponente j;  $\lambda_r^{m+1}$  [ $y^{-1}$ ] ist die Zerfallskonstante vom Mutternuklid m+1 des betrachteten Nuklids und  $\lambda_r^m$  [ $y^{-1}$ ] ist die Zerfallsrate des betrachteten Nuklids.  $I_i^m$  bezeichnet die gesamte Menge der Nuklide [mol] in der Komponente.

Abb. 2.1 zeigt das Beispiel eines einfachen Transportmodells auf der grafischen Benutzeroberfläche von AMBER. Es stellt einen Schadstofftransportprozess nach, bei dem Schadstoffe von einer Deponie durch Grundwasser und Flüsse bis zu einer Trinkwasserquelle gelangen.

Die Modellkomponenten bei AMBER unterscheiden sich in zwei Klassen: „Medium“ und „Transfer“. Im Beispiel von Abb. 2.1 gehören „Waste“ und „Aquifer“ zur „Medium“-Klasse und „Leaching“ sowie „Discharge“ zur „Transfer-Klasse“. Letztere beschreibt die Schadstofftransportrate zwischen den einzelnen Komponenten der „Medium“-Klasse. In der oben gezeigten Abb. 2.1 ist damit der Schadstofftransport zwischen „Waste“, „Aquifer“, „River“ und „Sink“ gemeint. Alle Klassen können vom Benutzer auf der grafischen Benutzeroberfläche bearbeitet werden.

Bei einem neuen Modell werden im ersten Schritt alle Medienkomponenten erstellt. Für jede wird die Anfangskonzentrationen der betrachteten Nuklide und deren Zerfallsbeziehungen festgelegt (Definition der Anfangswerte von  $I_i^m$  und der Zerfallskonstante  $\lambda_r^{m+1}$ ). Im oben angeführten Beispiel wurde die Schadstoffkonzentration nur für die erste Komponente (Waste) angegeben. Für die anderen Medium-Komponenten wurden die Anfangskonzentrationen der Nuklide auf null gesetzt.

Nachdem die Modellkomponenten und die Transferkomponenten vollständig erzeugt sind, definiert man ihre Parameter über ein Parameterfenster (Siehe Abb. 2.2). Darin werden Name, Beschreibung, Werte, statistische Verteilung, Verteilungsmaße usw. einzeln festgelegt. Parameter ohne festgelegten Wert, werden in dieser Arbeit „probabilistische Parametern“ genannt. Für die probabilistischen Parameter generiert AMBER Werte entsprechend Ihrer statistischen Verteilungen und setzt diese später zur Simulation ein.



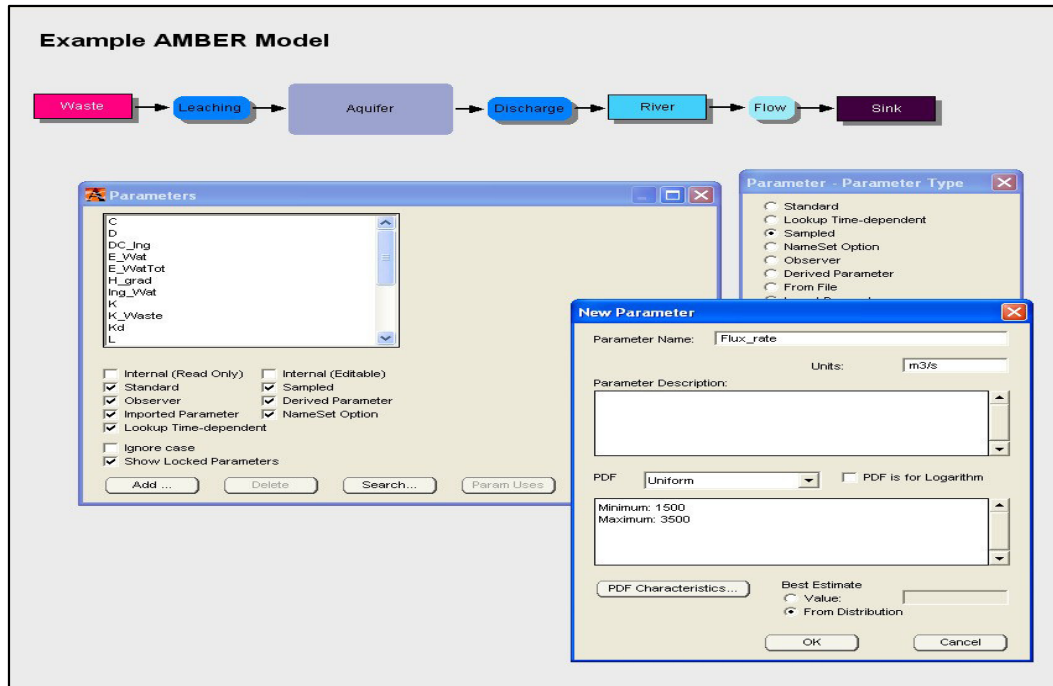


Abb. 2.2 Fenster zur Definition für die Parametereigenschaften im Modell /Bildschirmausschnitt der AMBER-GUI/

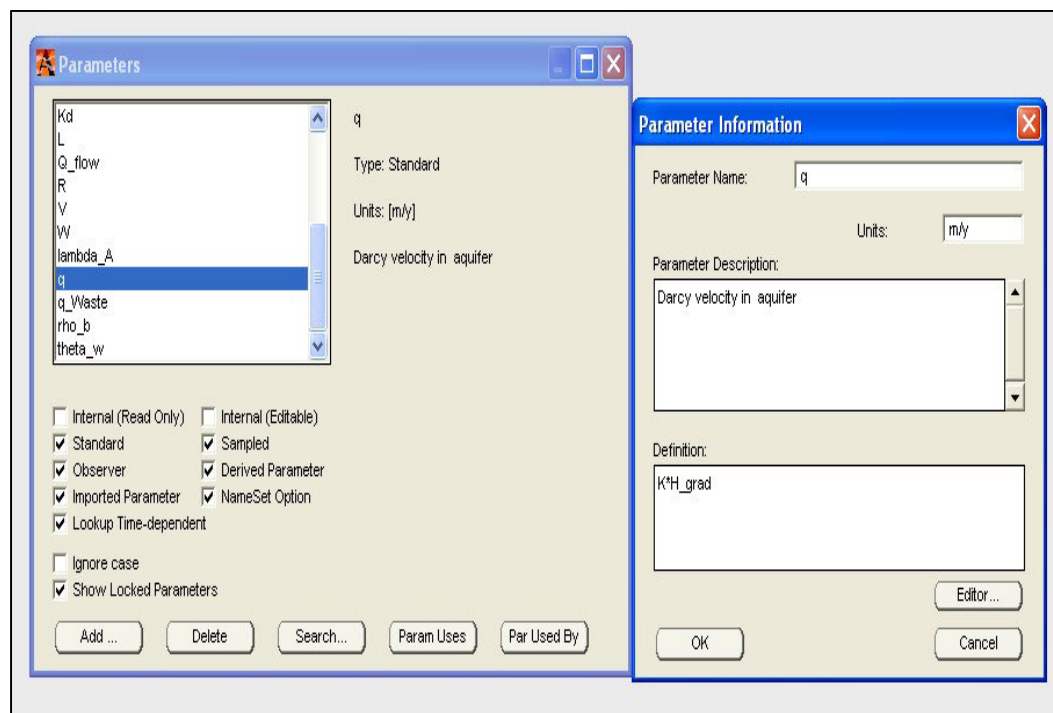


Abb. 2.3 Fenster zur Definition der mathematischen Beziehungen zwischen den Parametern /Bildschirmausschnitt der AMBER-GUI/

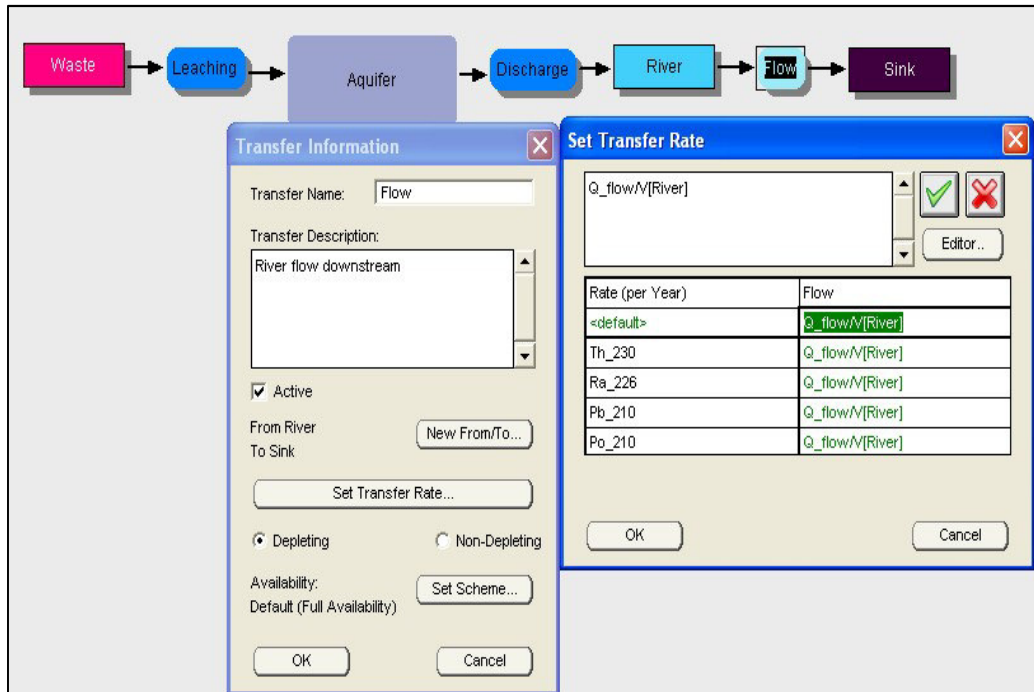


Abb. 2.4 Definitionsfenster für die Modellkomponenten /Bildschirmausschnitt der AMBER-GUI/

AMBER ermöglicht es dem Nutzer die mathematische Beziehung zwischen den Parametern auf der Benutzeroberfläche zu definieren (Siehe Abb. 2.3). Im nächsten Schritt werden alle probabilistischen und deterministischen Parameter festgelegt. Dies geschieht in den Definitionsfenstern von Abb. 2.2 und Abb. 2.3. Die Parameter können dann zu den vorher definierten Transportkomponenten in mathematische Beziehung gesetzt werden, beispielsweise zur Festlegung der Übergangsgeschwindigkeit zwischen den Medienkomponenten ( $\lambda_{ij}$  in Gleichung 2-1). Zulässig sind einfache Funktionen: z.B. Addition, Subtraktion, Multiplikation und Division usw. Wie weiter in Abb. 2.4 dargestellt, wird beim Doppelklick auf eine Transportkomponente ein Dialogfenster aufgerufen, in dem der Nutzer die advective Transportrate einzelner Radionuklide in Bezug auf Wasserfließrate (Q) definiert.

Die Transferkomponenten liefern die Transferrate zwischen den Medienkomponenten ( $\lambda_{ij}$  in Gleichung 2-1). Die Transfer-Rate kann durch eine vom Nutzer frei definierbare mathematische Gleichung berechnet werden. Z.B.

$$\lambda_{ij} = \frac{(P_1 - P_2) \times L \times H}{A} \quad (2-2)$$

wobei  $P_1$  [Pa] und  $P_2$  [Pa] den hydraulischen Druck auf den beiden Seiten der Komponente bezeichnen.  $L$  [m] ist die Pfadlänge,  $H$  bezeichnet die hydraulische Permeabilität des Transportmediums,  $A$  ist die Querfläche des Bezugsmediums. Alle genannten Parameter sind vom Benutzer festzulegen. Die genaue Vorgehensweise wird in den folgenden Schritten erläutert.

Wenn die Modellkomponenten und die dazugehörigen Parameter vollständig festgelegt sind, hat jede Medienkomponente eine eigene Gleichung 2-1. Die Transferkomponenten berechnen für Transferraten zudem weitere Gleichungen. Aus diesen Gleichungen ergibt sich ein Gleichungssystem, welches das Modell ausmacht.

AMBER hat zwei integrierte Solver (Löser) für die Kalkulation des entstandenen Gleichungssystems. Zum einen den „Laplace-Transformation-Solver“, der für Modelle mit zeitunabhängigen oder schrittweise zeitabhängigen Parametern genutzt wird. Die Parameterwerte sind hier über die Simulationszeit konstant oder ändern sich diskret nach den Rechenzeitschritten. Der andere Solver ist ein „Time-Step-Adaptive-Solver“, der für die Modelle mit zeitabhängigen Prozessen (z.B. veränderte Wasserfließrate, veränderte Transportparameter usw.) gebraucht wird. Die beiden Löser können nur für Modelle mit mathematisch vereinfachtem eindimensionalem Transport verwendet werden, was die Einsatzmöglichkeit und Rechenflexibilität vom AMBER begrenzt.

Nach den Simulationsläufen stellt AMBER die Ergebnisse grafisch als Diagramm dar. Die möglichen Ausgabearten sind (Siehe Abb. 2.5):

- Ergebnisläufe mit der Zeit
- Kumulative Verteilungsfunktion von Ergebnissen
- Streudiagramm (Scatterplot) der Ein- und Ausgabeparameter
- Statistische Quantile der Ergebnisgrößen

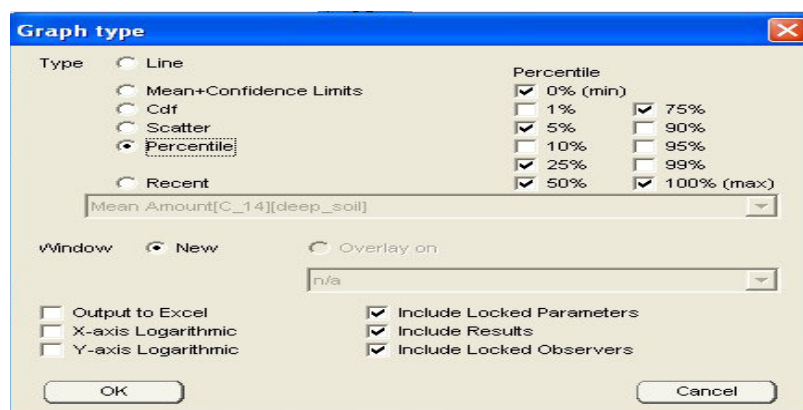


Abb. 2.5 Fenster der grafischen Darstellungsmöglichkeiten /Bildschirmarschnitt der AMBER-GUI/

Der Modellaufbau und alle Ein- und Ausgabeparameter können als ASCII-Datei gespeichert und von AMBER wieder geöffnet und bearbeitet werden.

## 2.2 GoldSim

GoldSim ist eine kommerzielle Software für die probabilistische Simulation im Ingenieursbereich sowie in der Forschung. Sie wurde von dem Unternehmen GoldSim Technology Group LLC entwickelt.

GoldSim hat ein Rechenmodul speziell für den Transport von Schadstoffen, das zur Simulation für eine Langzeitsicherheitsanalyse eines Endlagers genutzt werden kann [16].

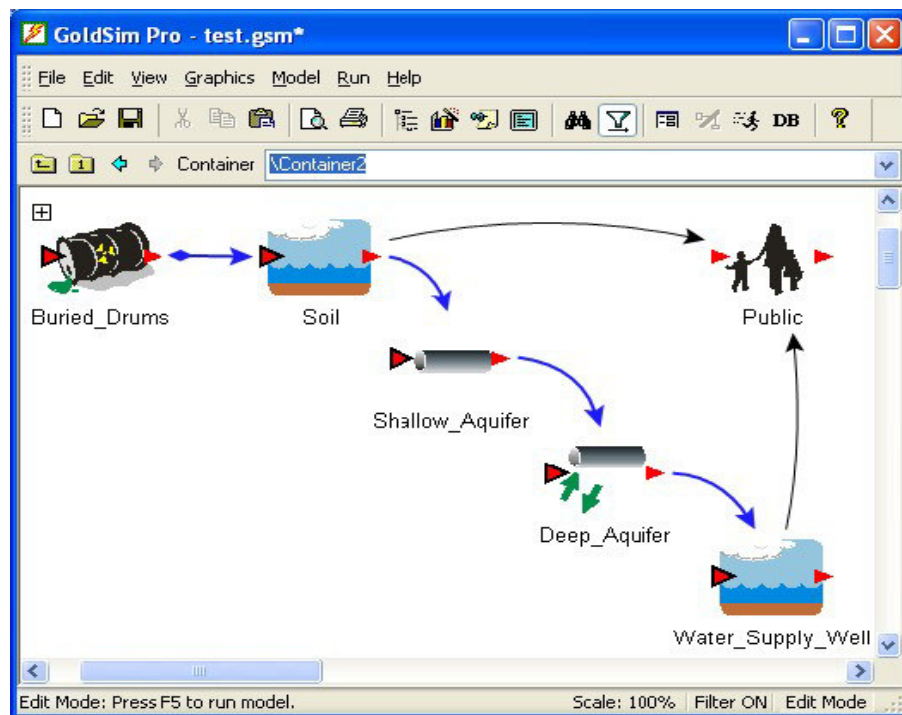


Abb. 2.6 Nutzeroberfläche von GoldSim. Hier werden die Komponenten eines Modells erstellt /Bildschirmausschnitt der GoldSim-GUI/

GoldSim hat eine grafische Benutzeroberfläche für den Modellaufbau (Siehe Abb. 2.6). Ähnlich wie bei AMBER werden die Analysemodelle aus einzelnen Komponenten zusammengesetzt, z.B. aus Schadstoffquelle, Umweltmedien, Transportwegen, Biosphäre usw. Jede Modellkomponente in GoldSim hat eine implementierte Transporteigenschaft und eigene Differenzialgleichungen.

Das Beispiel in Abb. 2.6 beinhaltet „Buried\_Drums“ (vergrabene Fässer) als Schadstoffquelle und „Public“ (die Allgemeinheit) als Biosphärenmodell. Eine Schadstoffquelle im Bereich der Kerntechnik bezieht sich auf Menge und Art der freigesetzten Radionuklide und wird Quellterm genannt. Der hier dargestellte Quellterm setzt Schadstoffe entsprechend einer einstellbaren Rate frei. Das Biosphärenmodell berechnet aus den Schadstoffkonzentration in den Umweltmedien Risiko- oder Dosiswerte für den Menschen.

GoldSim bietet zwei Typen von Transportkomponenten an: „Cell pathways“ (Soil und Water\_Supply\_Well in Abb. 2.6) und „Pipe pathways“ (Shallow\_Aquifer und Deep\_Aquifer in Abb. 2.6). Eine „Cell Pathways“-Komponente beschreibt den Schadstofftransport in gemischten Umweltmedien. In dieser Komponente sind Löslichkeits- und Zerfallseigenschaften der Schadstoffe modelliert. Das bedeutet die Konzentration der Schadstoffe im Medium kann beim Transport variieren. Außerdem kann ein Teil des Schadstoffs ausfallen oder anders herum wieder aufgelöst werden. Der Transport der Schadstoffe zwischen den Komponenten erfolgt durch vereinfachten advektiven und diffusiven Transport.

Für den advektiven Transport gilt:

$$f_{s,i-j} = c_{ims} * q. \quad (2-3)$$

Hier ist  $f_{s,i-j}$  die transportierte Schadstoffmenge von Komponente i zu Komponente j,  $c_{ims}$  die Schadstoffkonzentration in Komponente i und q die vom Benutzer festgelegte Wasserfließrate.

Für den diffusiven Transport gilt:

$$f_{s,i-j} = D_s (c_{ims} - \frac{c_{jns}}{K_{nms}}), \quad (2-4)$$

wobei  $f_{s,i-j}$  die diffusiv transportierte Schadstoffmenge von Komponente i zu Komponente j bezeichnet.  $D_s$  ist der Diffusionskoeffizient des entsprechenden Schadstoffs.  $c_{ims}$  ist die Schadstoffkonzentration in Komponente i und  $c_{jms}$  ist die Schadstoffkonzentration in der Komponente j.

Eine „Pipe Pathway“-Komponente beschreibt den detaillierten diffusiven und dispersiven Transport in porösen Medien. Die mathematische Funktion einer „Pipe Pathway“-Komponente ist:

$$Flux_s = (Q * c_{m,s} - (D_s + \alpha * Q) \frac{\partial c_{m,s}}{\partial x})|_{x=L} \quad (2-5)$$

wobei  $Flux_s$  der Konzentrationsfluss zur nächsten Komponente ist und Q die Fließrate der Flüssigkeit bezeichnet.  $c_{m,s}$  ist die Stoffkonzentration im Bezugsmedium,  $D_s$  ist der Diffusionskoeffizient,  $\alpha$  ist die Dispersionslänge des Schadstoffs im Medium und L ist die Länge des Bezugsmediums. Die Veränderung der Konzentration im Medium wird nach Gleichung 2-6 berechnet:

$$\frac{\partial c_{m,s}}{\partial t} = -[(\frac{Q}{n*A*R}) \frac{\partial c_{m,s}}{\partial x} - (\frac{\alpha*Q+D_s}{n*A*R}) \frac{\partial^2 c_{m,s}}{\partial x^2}] - c_{m,s} * \lambda_s \quad (2-6)$$

Hier bezeichnet n die Porosität des Mediums und A die Querschnittsfläche des Mediums. R ist der Retardationsfaktor für den Schadstoff,  $\lambda_s$  bezeichnet die Zerfallskonstante des Stoffs.

Ein „Pipe Pathway“ beschreibt den advektiven und dispersiven Transport durch Berechnung von Gl. 2-5 und Gl. 2-6 mittels Laplace-Transformation. Der Nutzer hat zudem die Möglichkeit

ein Transportnetz durch Kombination von „Pipe Pathway“-Komponenten (Siehe Abb. 2.7) zu erzeugen. Damit können komplexe Transportsysteme dargestellt werden.

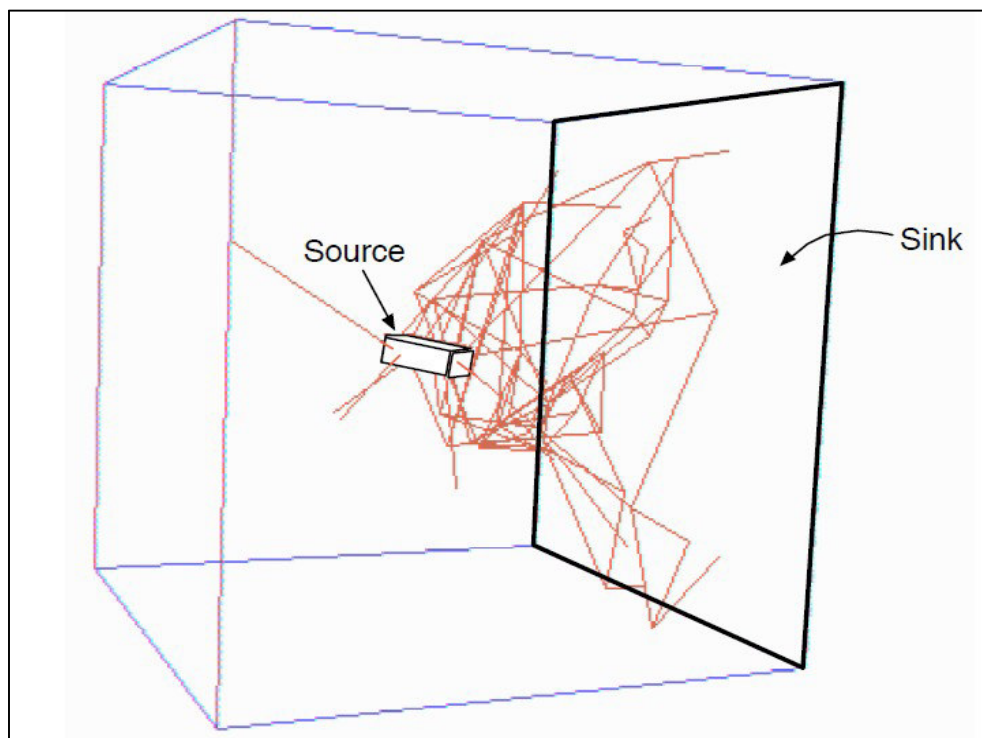


Abb. 2.7 Ein Beispiel für ein Kluft-Transportsystem, das durch Kombination von „Pipe Pathway“-Komponenten modelliert wird [16]

Der Nutzer kann durch Kombination der unterschiedlichen Modellkomponenten eine Modellstruktur herstellen. Jede Modellkomponente hat eine mathematische Gleichgewichtsgleichung. Aus der Modellstruktur wird dann ein mathematisches Gleichungssystem aus den Gleichungen der Modellkomponenten aufgebaut und im nächsten Schritt gelöst. Der Unterschied zwischen GoldSim und AMBER besteht in der Art der herangezogenen Gleichungen, AMBER führt die Simulation mit einer generellen linearen gewöhnlichen Differential-Transportgleichung für Modellkomponente und vom Benutzer frei definierbaren relativ einfachen mathematischen Gleichungen zwischen Parametern durch. GoldSim hingegen benutzt eigene Komponenten mit implementierten komplizierteren partiellen Differenzialgleichungen. GoldSim offeriert daher eine größere Rechenkomplexität gegenüber AMBER. AMBER bietet hingegen die Möglichkeit, dass die mathematischen Modelle frei verwaltet werden können. Nach der Simulation bietet GoldSim eine ähnliche Diagrammdarstellung wie AMBER für die Ergebnisse und die statistischen Maßzahlen der Ergebnisse.

GoldSim und AMBER sind kommerzielle Software-Produkte, die in ihrem Funktionalitätsumfang und ihrer Modellaufbautheorie vergleichbar sind. Aufgrund der ähnlichen Eigenschaften haben sie ähnliche Vor- und Nachteile.

Die wesentlichen Vorteile der kommerziellen Software-Produkte AMBER und GoldSim sind im folgendes aufgeführt:

- Sie haben eine mächtige grafische Benutzeroberfläche, auf der ohne Ausnahme alle Modellierungsprozesse durchgeführt werden können
- Sie bieten die Flexibilität eine beliebige Anzahl von Modellkomponenten in benutzerdefinierter Form zu kombinieren.
- Der Benutzer kann festlegen, welche Parameter als probabilistische Parameter mit bestimmten Verteilungen behandelt werden.
- Die Ergebnisse können automatisch grafisch dargestellt und analysiert werden.
- Die Ein- und Ausgabeparameter werden in textförmigen Modelldateien dokumentiert und können rekonstruiert werden.
- Die Software hat eine etablierte Qualitätssicherung.

In Hinblick auf die Nutzung für die Lehre und Forschung gibt es auch Nachteile bei der Verwendung der genannten Software-Programme:

- Die Komplexität des räumlichen und physikalischen Umfangs beider Programme ist begrenzt.
- Sie sind praktisch nur für den Schadstofftransport anwendbar. Die Situation im Endlagersystem mit komplizierter Geometrie und inhomogenen Materialien und anderen THMC-Prozessen kann mit ihnen nicht modelliert werden.

## 2.3 ALLIANCES

ALLIANCES ist eine Software-Plattform für die Modellierung und Analyse für die Langzeitsicherheitsanalyse von Endlagern radioaktiver Abfälle. Sie wurde von „Commissariat à l'énergie atomique et aux énergies alternatives“ (CEA, auf Deutsch „Französische Kommissariat für Atomenergie und alternative Energien“) und „Agence Nationale pour la Gestion des Déchets Radioactifs“ (ANDRA, auf Deutsch „Nationale Agentur für das Management radioaktiver Abfälle“) entwickelt [14].

Folgende Eigenschaften dieser Software sind hervorzuheben:

- Die Möglichkeit einer Einbindung bereits existierender numerischer Codes als Modellkomponenten und als Modellketten mit Datentransformation zwischen den einzelnen Komponenten.
- Ihre Eignung für die probabilistische Analyse von Parametern

In einem wesentlichen Punkt besteht ein Unterschied zwischen dem von Forschungseinrichtungen entwickelten ALLIANCES-Programm und den vorgestellten kommerziellen Software-Produkten: Ersteres nutzt keine eigenen integrierten Solver, sondern ruft externes numerisches Programm auf. Das ermöglicht unter anderem die Anwendung der Finite-Elemente-Methode sowie der Finite-Differenzen-Methode. Dadurch beherrscht ALLIANCES die

Modellierung thermisch, hydraulisch, mechanisch und chemisch gekoppelter Prozesse, die auf den genannten kommerziellen Software-Produkten nicht ohne weiteres modelliert werden können.

ALLIANCES hat das „Mesh and Field Data Model“ („MED“). Dies ist ein spezielles Datenmodell für räumliche Modell- und Meshing-Daten im sogenannten MED-Format. Dieses Format wurde von „Französische Kommissariat für Atomenergie und alternative Energien“ (CEA) und „Électricité de France“ (EDF) beim „Salome Platform“-Projekt entwickelt. Es ermöglicht Fremdcodes den Zugriff auf die Modellgeometriedaten. ALLIANCES hat mit der Salome-Plattform das MED-Dateiformat gemein und kann die dort gesammelten räumlichen Modellinformationen bearbeiten und darstellen. (siehe Abb. 2.8).

ALLIANCES hat eine Benutzeroberfläche, die auf der Python-Skript Sprache [18] und Qt-Designer [17] basiert.

Der zentrale Gedanke von ALLIANCES ist die Einbindung externer numerischer Simulationscodes zur Berechnung auf einer einzigen Software-Plattform. Die eingebrachten Codes stehen in den Kästchen in Abb. 2.8 und betreffen verschiedene Bereiche: PORFLOW und CAST3M sind für die hydraulische Modellierung; MT3D, TRACES sind für die Stofftransportmodellierung; PHREEQC und CHESS dienen der chemischen Modellierung; ASTER betrifft die Modellierung für thermisch, hydraulisch und mechanisch gekoppelte Mechanismen. Die Codes können von der ALLIANCES-Plattform aufgerufen werden um die probabilistischen Simulationen mit zufällig gesampelten Parametern durchzuführen. ALLIANCES bietet darüber hinaus die Möglichkeit, Code-Komponenten durch Datentransformation zwischen den Codes zu koppeln. Auf diese Weise kann das Modell gekoppelte Prozesse abbilden. Beispielsweise wird für die Nuklidtransportmodellierung die Kombination von MT3D und PORFLOW verwendet und die von PORFLOW berechneten Daten (Fließgeschwindigkeit usw.) werden zu MT3D transformiert um die Stoffemissionsprozesse zu modellieren.

Die Bestandteile der Software sind in der Abb. 2.8 als Komponenten aufgeführt. Die gekoppelten Modelle können auch mit statistischen Parametern probabilistisch analysiert werden. Die eingebundenen Codes müssen kompatibel zum Meshing-Format sein, dessen Konvertierungsmethode nicht für jeden fremden numerischen Code verwendbar ist.



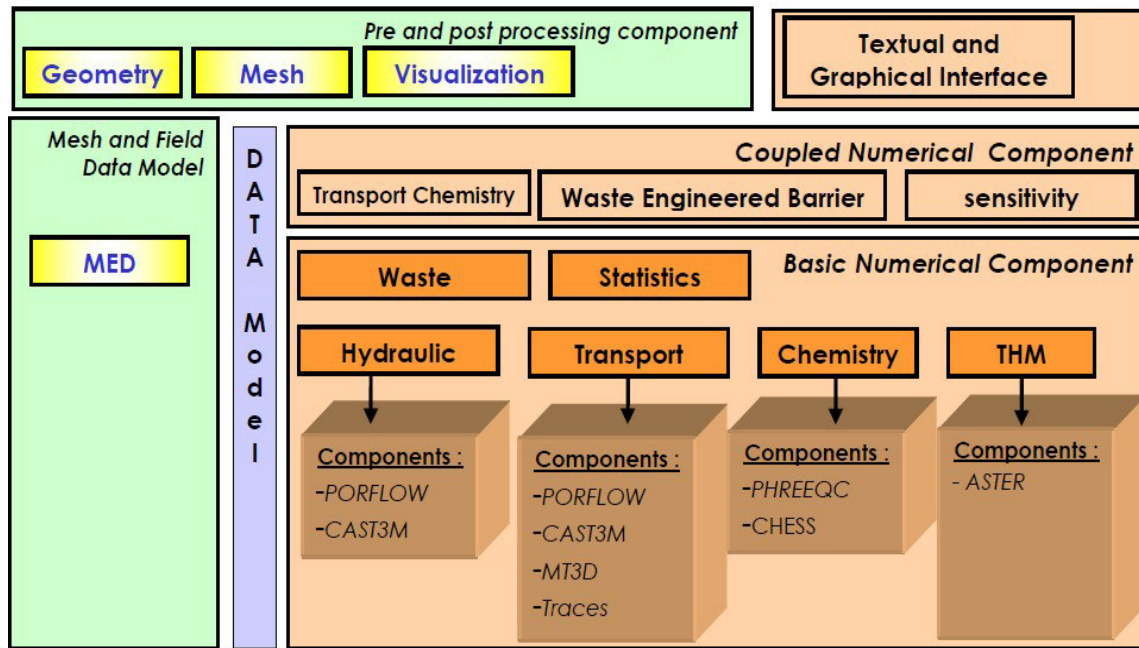


Abb. 2.8 Multi-Ebenen-Kopplungsarchitektur von ALLIANCES [14]

ALLIANCES hat den zentralen Vorteil, dass es die externen numerischen Codes mit Datentransformation funktionsfähig einbinden kann und eine probabilistische Analyse von Prozessmodellen ermöglicht. Es gibt aber auch Einschränkungen bei der Nutzung der Plattform. Die Wesentlichen sind im Folgenden aufgeführt:

- Modellkomponenten, die gekoppelt werden sollen, müssen eine gemeinsame Modellgeometrie haben. Dies verhindert die Einbindung von integrierten Modellen mit unterschiedlichen Geometriebereichen (z.B. Nahfeld, Fernfeld, Biosphäre usw.) für die Beschreibung der Endlagersysteme. Die Auswahl an externen Codekomponenten ist begrenzt, da die Plattform von wenigen Organisationen für wenige spezielle Projekte entwickelt wurde. Die oben erwähnte Meshingmethode muss mit den einzubringenden Fremdcodes kompatibel sein, was die Flexibilität im Sinne einer Einbindung anderer Codes begrenzt.
- Die Benutzerfreundlichkeit der teils grafischen und teils Python-textbasierten Umgebung ist begrenzt, was der Anwendungsumfang der Plattform insbesondere bei der Lehre einschränkt.
- ALLIANCES ist in geringerem Umfang qualitätssichert als die oben erwähnten kommerziellen Codes.

Es gibt noch andere Rechenprogramme, die bei der Sicherheitsanalyse für Endlager häufig angewendet werden. Das Rechenprogramm MARNIE [19] wurde im Auftrag des Bundesministeriums für Umwelt, Naturschutz und Reaktorsicherheit (BMU) von der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH entwickelt. MARNIE ist ein Rechenprogramm zur Simulation eindimensionaler Transportprozesse und berechnet den Laugen – und Radionuklidstrom in einem Endlagerbergwerk. Das Programmpaket EMOS [20], das von der GRS mit Mitteln des Bundesministeriums für Bildung, Wissenschaft, Forschung und

Technologie (BMBF) entwickelt wurde, ist ein Instrumentarium zur Analyse der Langzeitsicherheit von Endlagern. Die beiden Rechenprogramme werden an dieser Stelle nicht vertieft beschrieben.

### **3 Einführung in die Software-Plattform RESUS**

Die am Institut für Endlagerforschung gesammelten Erfahrungen mit der heutzutage vorhandenen Simulationssoftware haben gezeigt, dass für die Anwendung bei Lehre und Forschung, speziell im Bereich der Endlagerforschung, die Benutzerfreundlichkeit und Anwendernähe der Software noch nicht im gewünschten Maße vorhanden ist.

Wie in Kapitel 1 erläutert, gibt es zwei Typen von häufig anzuwendenden Analysemodellen in der Endlagerforschung. Diese sind: integrierte Modelle und Prozessmodelle. Letztere beschreiben detailliert die THMC-gekoppelten Prozesse in der Betriebs- und Nachbetriebsphase eines Endlagers. Sie können aufgrund der komplexen Strukturen eines Endlagersystems im geometrischen und physikalischen Bereich normalerweise nur einen Teil des gesamten Endlagersystems abbilden.

Im Gegensatz zu den Prozessmodellen betrachten die integrierten Modelle das Endlager als ein gesamtes System, wodurch erst die Bewertung der Sicherheit des Endlagersystems ermöglicht wird. Üblicherweise bestehen integrierte Modelle aus Modellkomponenten, welche jeweils einen Teil eines Endlagersystems vereinfachend beschreiben. Die Herausforderung bei der probabilistischen Analyse und der Sensitivitätsanalyse der Modelle in der Endlagerforschung mit integrierten Modellen besteht darin, die sehr komplexen THMC-gekoppelten Prozessmodelle als Modellkomponenten zu implementieren. Gelingt dies, so sind mit den integrierten Modellen sehr realitätsnahe Simulationen durchführbar. Natürlich müssen trotz steigender Rechenkapazitäten der Computertechnik auf Grund des hohen Rechenaufwands bei dem Modellaufbau immer Vereinfachungen oder Änderung in der probabilistischen Modellierung durchgeführt werden. Die Modellstruktur wird so lange angepasst, bis der Rechenaufwand für die probabilistische Modellanalyse geeignet ist. Dabei ist zu beachten, dass Modellvereinfachungen immer zu rechtfertigen sind.

Die Softwareplattform RESUS wird mit den vier folgenden Anforderungen entwickelt:

- Sie soll die probabilistische Analyse und die Sensitivitätsanalyse mit integrierten und Prozessmodellen im Endlagerforschungsbereich automatisieren und die Rechnungen nachvollziehbar dokumentieren.
- Sie muss eine Simulation von THMC-gekoppelten Vorgängen mit existierenden fremden THMC-Codes durchführen können.
- Sie soll eine Plattform bieten, die zur Integration von Prozessmodellen mittels Datentransformation zu einem integrierten Modell dient.

- Die Softwareplattform soll zur Lehre, professionellen Ausbildung und Forschung verwendet werden und deshalb eine hohe Benutzerfreundlichkeit mit einer grafischen Benutzeroberfläche aufweisen.

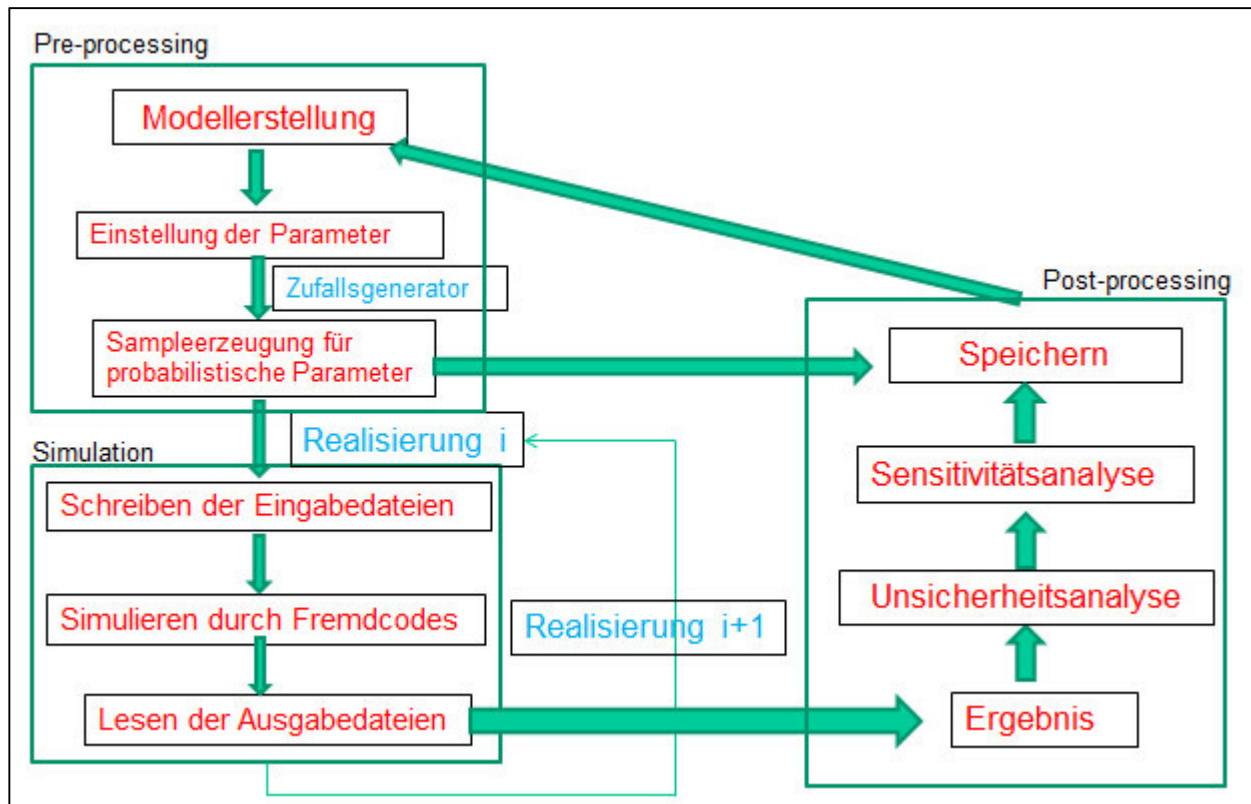


Abb. 3.1 Arbeitsprozedur von RESUS

Abb. 3.1 zeigt die Arbeitsweise von RESUS ausgehend von der Modellerstellung bis zur Modellanalyse. Ähnlich wie andere Simulationssoftware arbeitet RESUS in drei Hauptschritten: Pre-Processing, Simulation-Processing und Post-Processing. Als erster Schritt im Pre-Processing wird das zu rechnende Modell initialisiert. Sowohl einzelne Modelle wie auch gekoppelte Modellstrukturen, die das Endlagersystem abbilden, können von RESUS verarbeitet werden.

Für die Modellierung muss man Parameter verwenden. In der probabilistischen Modellanalyse werden zwei Parameter-Typen unterschieden: deterministische und probabilistische Parameter. Deterministisch sind Parameter, die als konstante Werte in die Simulationen eingefügt werden. Probabilistische Parameter hingegen haben Werte, die durch eine Messung oder die Szenarienprognose nicht genau ermittelt werden können. Folglich werden die probabilistischen Parameter bei der Simulation als „unsichere“ Parameter mit zufallsverteilten variablen Werten behandelt.

RESUS wird als Softwarewerkzeug in Lehre, Ausbildung und Forschung verwendet. Die angesprochene Benutzergruppe ist daher sehr inhomogen. Um den verschiedenen Ansprüchen hinsichtlich Benutzerfreundlichkeit und Benutzbarkeit gerecht zu werden, sind zwei Methoden

der Modellerstellung in RESUS implementiert worden: Die Benutzeroberfläche-basierte (GUI-basierte) Methode und die dateibasierte Methode.

Mit der GUI-basierten Methode sind ausschließlich die in RESUS fest implementierten Modelle simulierbar. Bei einer Analyse mit der GUI-basierten Methode erstellt RESUS automatisch anhand der Benutzereingaben auf der grafischen Benutzeroberfläche die so genannten Eingabedateien für die anzuwendenden Fremdcodes (siehe Abb. 3.2). Die Modellstruktur und die dazugehörigen probabilistischen und deterministischen Parameter werden durch die GUI-basierte Modellerstellungsmethode in der RESUS-GUI eingestellt. Der Nutzer kann auf der RESUS-GUI weiterhin die Parameterwerte des Modells eingeben. Der ganze Prozess der GUI-basierten Methode erfolgt ausschließlich auf der grafischen Benutzeroberfläche.

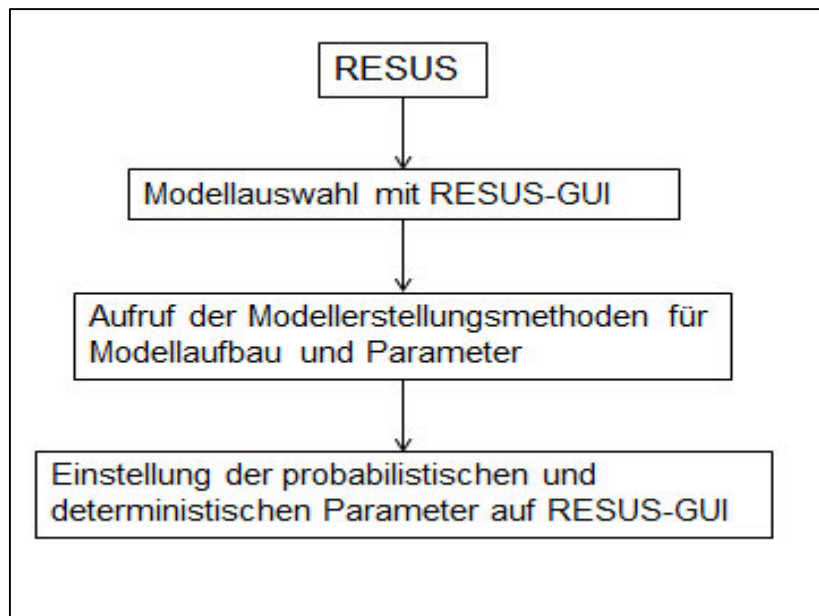


Abb. 3.2 Arbeitsprozeduren mit der GUI basierten Methode.

Diese Abbildung ist die genauere Beschreibung des „Pre-Processing“ Vorgangs aus Abb. 3.1 unter Verwendung der GUI-basierten Methode. Analysemodelle sind mit der GUI-basierten Methode einfacher und schneller herzustellen als mit der dateibasierten Methode. Doch ist die GUI-basierte Methode auch Einschränkungen unterworfen bezüglich Umfang, Komplexität und Flexibilität der Modelle. Deswegen werden Modelle mit dieser Methode vor allem als Rechenbeispiele bei der Lehre und Ausbildung eingesetzt. Die genaue Vorstellung der Modellerstellung mittels der GUI-basierten Methode wird in Kapitel 4 behandelt.

Neben der GUI-basierten Modellerstellungsmethode ist die dateibasierte Methode das zweite Modellherstellungsverfahren bei RESUS. Diese ist stärker auf den Einsatz in der Forschung ausgerichtet und arbeitet mit einer sogenannten Beschreibungsdatei und entsprechenden Template-Eingabedateien. Bei der Arbeit mit der dateibasierten Methode muss der Nutzer zuerst eine RESUS-Beschreibungsdatei erzeugen, wo er die Grundinformation über die Modellstruktur und die dazugehörigen Fremdcodes hinterlegt. Mit dieser Information sucht RESUS die ebenfalls vom Nutzer vorbereiteten Template-Eingabedateien der einzelnen

Modellkomponenten und liest die Eingabedateien jeder Modellkomponente durch. RESUS erkennt die in den Template-Eingabedateien gekennzeichneten probabilistischen Parameter automatisch. Die eingelesene Information wird zur RESUS-GUI und weiter zur Simulation und Analyse geleitet.

Die genaue Vorgehensweise der dateibasierten Methode wird in Kapitel 4.5 vorgestellt.

Bei einem Vergleich der Modellerstellungsmethoden weist die dateibasierte Methode gegenüber der GUI-basierten Methode große Flexibilitätsvorteile auf: Es lassen sich dateibasiert ohne Einschränkungen beliebige Formen von Modellen mit Template-Eingabedateien aus Fremdcodes erzeugen. Dies erfordert allerdings erhebliche Vorkenntnisse des Nutzers hinsichtlich der angewendeten Fremdcodes. Deshalb findet die dateibasierte Methode vornehmlich in Wissenschaft und Forschung ihre Anwendung. Die dateibasierte Modellerstellungsmethode ist in dieser Version von RESUS nur einsetzbar, wenn die Eingabedateien der numerischen Fremdcodes textförmig (ASCII) sind. Bei binären Eingabedateien ist die Identifikation und Manipulation der Parameter problematisch.

Nach der Modellerstellung mit einer der beiden Methoden müssen alle benötigten Parameter zur Modellstruktur hinzugefügt werden. Die deterministischen Parameterwerte werden vom Nutzer direkt festgelegt. Bei den probabilistischen Parametern muss der Nutzer zunächst die Wahrscheinlichkeitsverteilungen spezifizieren. Die statistischen Verteilungen werden normalerweise aus Messdaten und Expertenmeinungen [21] abgeleitet. Nach diesen werden Zufallszahlen als Parameterwerte generiert.

Sind alle Parameterwerte erstellt, kann das Modell berechnet werden. Die Softwareplattform RESUS hat keine eigenen Solver für die Modellrechnung, sondern kalkuliert die Modelle durch den Aufruf externer Fremdcodes. Diese benötigen für die Simulationsrechnung Eingabedateien, welche durch RESUS automatisch aus den Parameterdaten erstellt werden. Bei der probabilistischen Simulation geschieht dies für jeden Durchlauf mit neu gesampelten Parameterwerten. Wenn also Eingabedateien vorliegen, so können die externen numerischen Codes die Simulationsrechnung durchführen. Die errechneten Daten werden als Ausgabedateien an RESUS übergeben. RESUS speichert die Rechenergebnisse als so genannte „ResultsParameter“. Dieser gesamte Prozessablauf der Modellrechnung wird in RESUS vollautomatisch durchgeführt und benötigt keine weiteren Benutzereingaben.

RESUS ermöglicht die Kombination von mehreren Modellkomponenten. Jede Modellkomponente stellt ein sogenanntes „Submodell“ dar, das Teil des gesamten Modells ist. Der Zusammenschluss mehrerer Submodelle ergibt eine sogenannte Modellkette, wie in Abb. 3.3 gezeigt. Zur besseren Übersichtlichkeit werden Modellkomponenten im weiteren Verlauf dieser Arbeit „Submodelle“ genannt.

Jedes von RESUS erzeugte Submodell benötigt eigene Eingabedateien und wird von einem Fremdcode berechnet. Die errechneten Daten werden auf RESUS eingelesen und zur nächsten Modellkomponente in der Modellkette als zeitabhängige Eingabedaten weitergeleitet. Bei einer Simulationsrechnung wird die Modellkette durchgegangen und die entsprechenden Zwischen- und Endergebnisse werden gespeichert. Der gesamte Austausch der Daten zwischen RESUS

und den Submodellen von RESUS wird durch I/O-Stream realisiert. Ebenso wird auf die Speicherdateien des Modells mit I/O-Stream-Methoden zugegriffen.

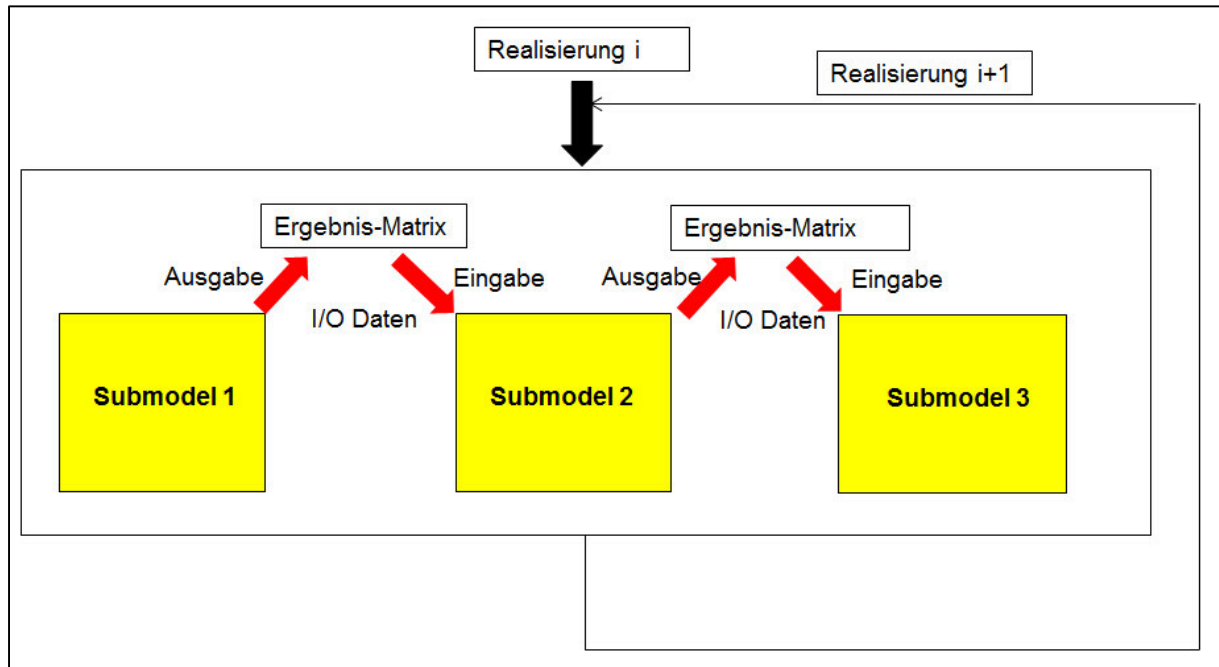


Abb. 3.3 Modellkette eines Simulationsmodells bei RESUS

Wenn die Modellrechnungen abgeschlossen sind, kann das Post-Processing beginnen. Das Post-Processing fokussiert sich auf die Darstellung und Analyse der Rechenergebnisse. RESUS bietet die Möglichkeit, die zeitabhängigen Ergebnisdaten zu plotten, logarithmisch zu transformieren, sowie nach Aufbereitung im Histogramm und Scatterplot darzustellen. Das gesamte Post-Processing wird auf der RESUS-GUI vollautomatisch durchgeführt.

Im folgenden Kapitel 4 werden die Modellierungs- und Analyseprozesse mit RESUS anhand mehrerer Beispiele im Detail vorgestellt. Das darauf folgende Kapitel 5 beleuchtet den Code-Aufbau von RESUS. Der Quellcode von RESUS hat eine objektorientierte Form, die besonders gut zu der hier erörterten modularen Struktur der Modelle passt.

## 4 Vorstellung der Funktionalitäten von RESUS (Nutzeranleitung)

RESUS ist eine Simulations- und Analyseplattform, die die meisten Prozesse von der Modellherstellung bis zur Ergebnisanalyse mit einer graphischen Benutzeroberfläche verwaltet. Dieses Kapitel behandelt die Demonstration der grafischen Benutzeroberfläche mit Beispielen aus Modellrechnung und Analyse. Es soll explizit auch als Anleitung für die Benutzung von RESUS dienen.

RESUS arbeitet nach dem in Abb. 3.1 dargestellten Arbeitsablauf. Bei Anwendung der GUI-basierten Methode wird jeder Arbeitsvorgang ausschließlich durch die RESUS-GUI gesteuert. Bei der dateibasierten Methode arbeitet der Nutzer erst dann mit der GUI, nachdem die Beschreibungs- und Template-Eingabedateien erstellt worden sind.

RESUS funktioniert unter MS-Windows-Betriebssystem Version: XP/Vista/7/8 x64 oder x86. Ein C++ Compiler wird für die Übersetzung des RESUS-Quellcodes vorausgesetzt. Eine Anforderung an die Rechner-Hardware ist nicht vorgesehen. Beim Aufruf der kompilierten RESUS-Anwendung „RESUS.exe“ erscheint die Benutzeroberfläche. Die Start-Benutzeroberfläche von RESUS sieht wie in Abb. 4.1 aus.

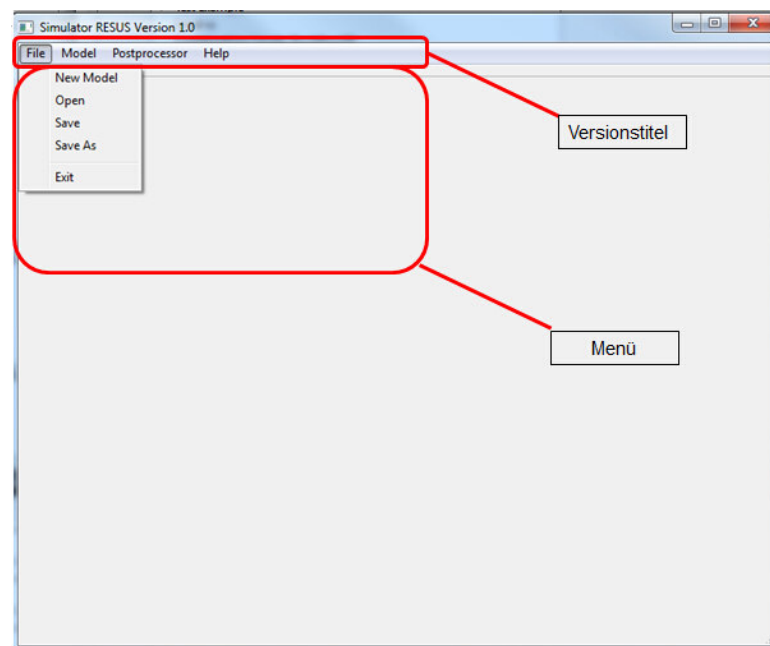


Abb. 4.1 Anfangsphase bei Aufruf der RESUS-GUI

Mit einem Mausklick an „New Model“ wird ein Steuerelement mit Reiteraufbau (Tabs) für die Einstellung, Berechnung und Analyse der Modelle aufgerufen, wobei die Reihenfolge der Tabs von links nach rechts nach dem Arbeitsprozess sortiert ist (Siehe Abb. 4.2).

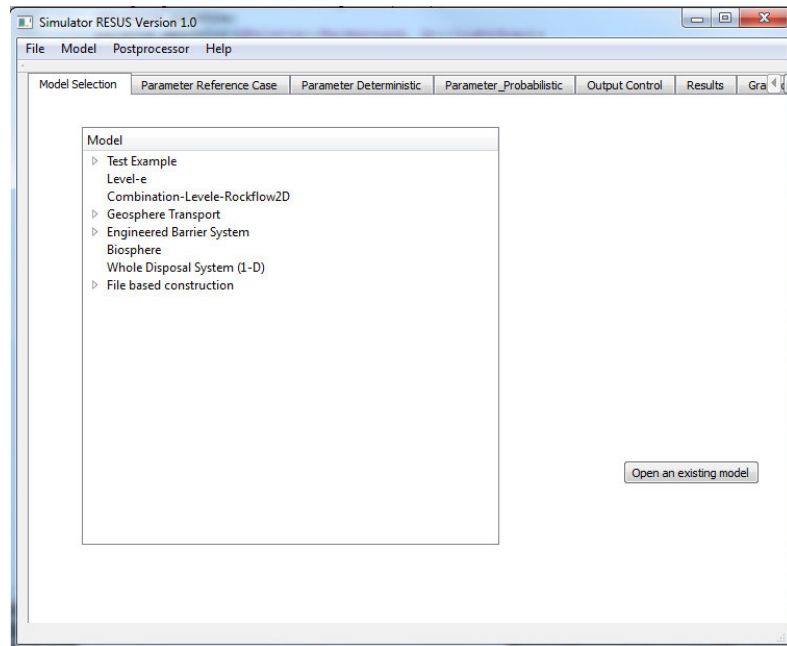


Abb. 4.2 Tabs-Aufbau der RESUS-Oberfläche

In RESUS gibt es folgende Tabs: „Model Selection“ für die Auswahl des anzuwendenden Modells; „Parameter Reference Case“ für die deterministische Festlegung der probabilistischen Parameter (probabilistische Parameter müssen grundsätzlich auch deterministische Referenzwerte haben, da ein deterministischer Testlauf bei der probabilistischen Modellrechnung mit RESUS durchgeführt werden muss); „Parameter Deterministic“ für die Einstellung der deterministischen Parameter; „Parameter\_Probabilistic“ für die Erzeugung der probabilistischen Parameterwerte entsprechend ihrer Verteilungen; „Output Control“ für die Auswahl nach gewünschten Ausgabeparametern, „Results“ für die Wertdarstellung der Rechenergebnisse (dieser Tab ist für eine tabellarische Darstellung der zeitabhängigen Rechenergebnisse vorgesehen, aber bleibt in dieser Version von RESUS noch ungenutzt); „Graphic Plot“ für die grafische Darstellung der zeitabhängigen Rechenwerte; „Statistic Display“ für die statistische Analyse und die Sensitivitätsanalyse. Die Benutzeroberfläche und die Tabs werden in den folgenden Kapiteln mit Modellbeispielen ausführlicher vorgestellt.



## 4.1 Ishigami-Modell in RESUS

Das Ishigami-Modell ist unser erstes Beispielmmodell für die Benutzung der GUI-basierten Methode. 3000 Durchläufe einer einfachen Modellgleichung sollen mit probabilistischen Parameterwerten gerechnet werden. Die daraus gewonnenen Ergebnisse werden schließlich in der Histogramm- und Scatterplotdarstellung (Streudiagramm) analysiert.

Das Ishigami-Modell ist ein Testmodell für die Sensitivitätsanalyse, das im Rahmen von RESUS besonders in der Lehre zur Sensitivitätsanalyse verwendet werden kann. Es besteht aus einer Gleichung mit vier probabilistischen Eingabeparametern und einem Ausgabeparameter. Gleichung 4-1 ist die Ishigami-Modellgleichung: [22][23]

$$Y = \sin x_1 + 7 \sin^2 x_2 + 0.1 x_3^4 \sin x_1 \quad (4-1)$$

Hier sind  $X_1$ ,  $X_2$ ,  $X_3$  und  $X_4$  probabilistische Eingangsparameter, die alle zwischen  $-\pi$  und  $\pi$  gleichverteilt sind.  $X_1$ ,  $X_2$  und  $X_3$  werden in der Gleichung einbezogen,  $X_4$  dient lediglich als Dummy-Parameter. Die vier Eingangsparameter sind voneinander unabhängig.  $Y$  ist der einzige Ausgabeparameter.

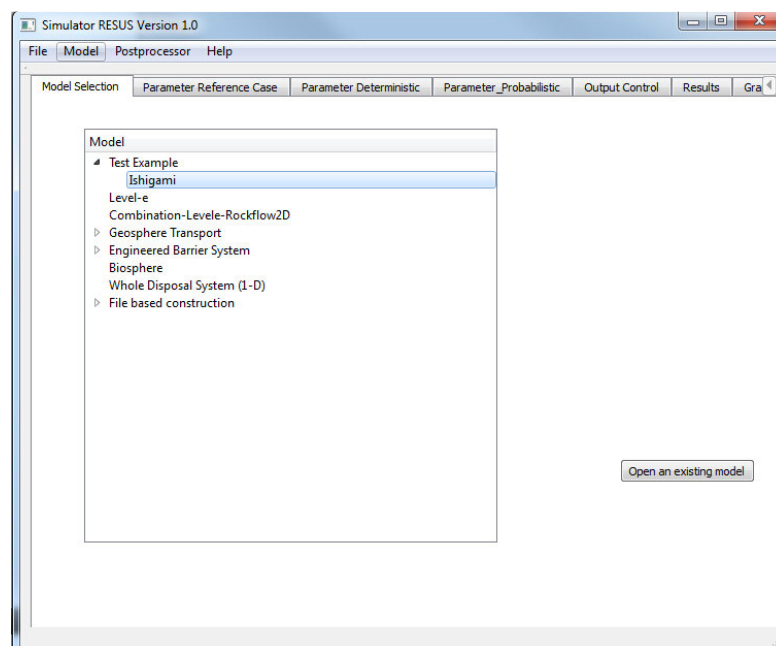


Abb. 4.3 Auswahl des Modells „Ishigami“ bei dem ersten Menü-Tab

Sobald ein Modell ausgewählt ist, springt die GUI zum zweiten Tab „Parameter Reference Case“ (siehe Abb. 4.4). Dort ist eine Tabelle zu sehen, mit den Spalten „Unit“ (Einheit), „Description“ (Beschreibung) und „Value Interval“ (physikalisches Wertintervall, aber bei Ishigami-Modell hat es keine physikalische Bedeutung). Die darin eingetragenen Daten sind lediglich als Hinweis für den Benutzer aufgeführt und können nicht geändert werden.

Wie bei diesem Modell üblich, gibt es vier Parameter  $X_1$ ,  $X_2$ ,  $X_3$  und  $X_4$ . Diese haben das selbe Wertebereich  $[-\pi, \pi]$  und sind ohne Einheit. Die Wertebereiche der Parameter sind in der Tabelle des „Parameter Reference Case“-Tabs eingetragen. Für jede Berechnung eines Modells ist die Festlegung genau eines Referenzparameterwert-Satzes erforderlich. Für diese Eingabe fixer Parameter wird die Zeile „Value“ genutzt. Die oberen drei Zeilen in der Tabelle (Unit, Description, Value interval) sind nicht editierbar, beim Anklicken erscheint eine Fehlermeldung (siehe Abb. 4.5).

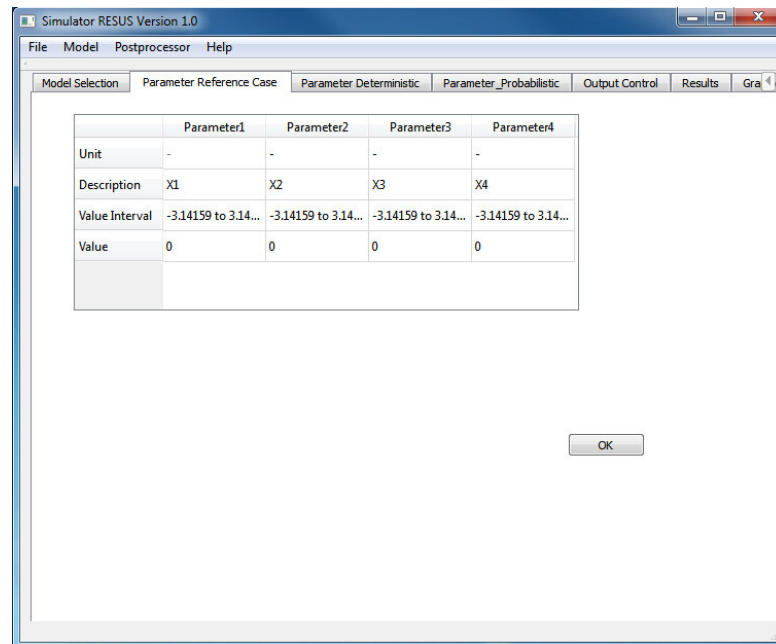


Abb. 4.4 Tabelle für die Festlegung der Referenzwerte der probabilistischen Parameter

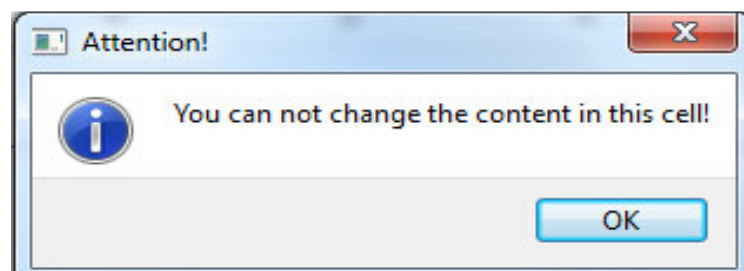


Abb. 4.5 Warnungsfenster beim Anklicken der Tabellenzelle „Description“

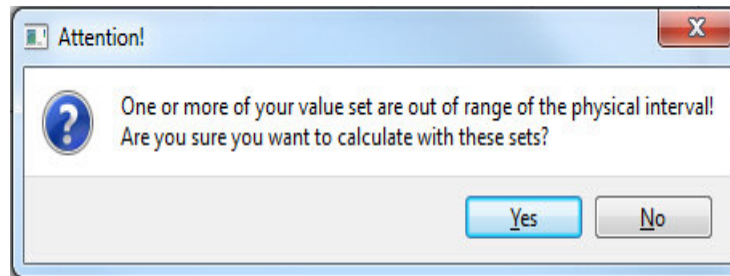


Abb. 4.6 Dialogfenster bei der Eintragung eines Parameterwertes außerhalb des gültigen Wertebereichs

In der „Value“ Zeile muss wie oben erläutert ein Referenzwert eingefügt werden. Wenn die Werteingabe für alle Parameter abgeschlossen ist, schließt der Benutzer die Eingaberoutine mit Klick auf „OK“. Gleichzeitig werden die eingetragenen Werte auf Konsistenz mit den Wertebereichen der Parameter geprüft. Sollte ein gegebener Wert außerhalb des Intervalls liegen, erscheint ein Dialogfenster wie in Abb. 4.6. Wird darin „ja“ gewählt, so werden die Werte dennoch beibehalten. Wählt der Nutzer „No“, so zeigt die GUI erneut die Eingabetabelle im Tab „Parameter Reference Case“.

Da im Ishigami-Modell nur die vier probabilistische Parameter existieren, überspringt RESUS den Einstellungsprozess der deterministischen Parameter. Die RESUS-GUI geht im nächsten Schritt direkt zum Tab „Parameter\_Probabilistic“, in dem die entsprechend ihrer Verteilung zufallsgenerierten Parameterwerte aufgeführt werden. Der Tab ist in Abb. 4.7 gezeigt. Links befindet sich eine Tabelle, in der die Namen, Wertebereich und „More Details“ für die Weiterbearbeitung der 3 Parameterwerte aus unserem Beispiel automatisch von RESUS eingetragen werden. Die Tabelle und der Knopf „Generate“ sind ausgegraut, solange keine Einstellung im Fenster „Sampling Size“ erfolgt.

Im nächsten Schritt werden Zufallswerte für das Modell festgelegt. Die Festlegung der Anzahl zu eventuellen Simulationsrealisierungen erfolgt im Dialogfenster „Sampling Size“ rechts in Abb. 4.7. Bei unserem Ishigami-Beispielmodell wollen wir 3000 Realisierungen verwenden. Wenn die Sample-Größe festgelegt ist, wird die Tabelle für die Einstellung der statistischen Eigenschaften der probabilistischen Parameter aktiviert. Diese Tabelle hat so viele Spalten wie probabilistische Parameter existieren. In der dritten Reihe „More Details“ erfolgt die Einstellung der Verteilungsparameter für den einzelnen Parameter (siehe Abb. 4.8).

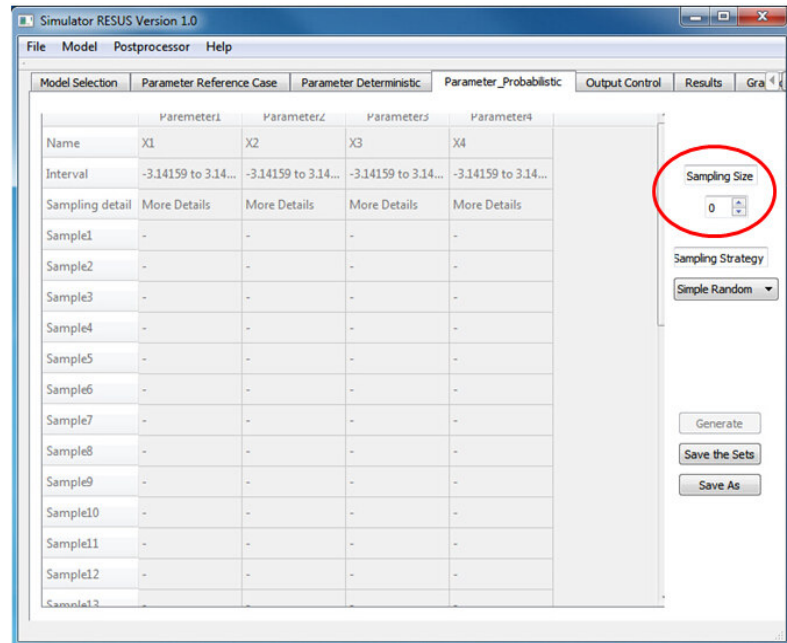


Abb. 4.7 RESUS-Tab für die Einstellung der entsprechend ihrer Verteilung zufallsgenerierten Parameterwerte der probabilistischen Parameter

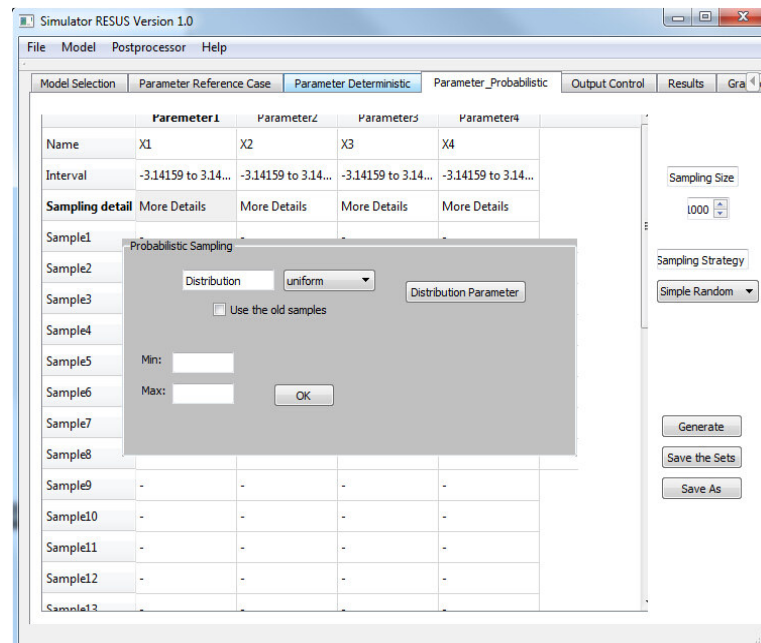


Abb. 4.8a Fenster für die Einstellung der Verteilungsfunktion eines probabilistischen Parameters bei der Wahl der Verteilungen „uniform“ und „log\_uniform“

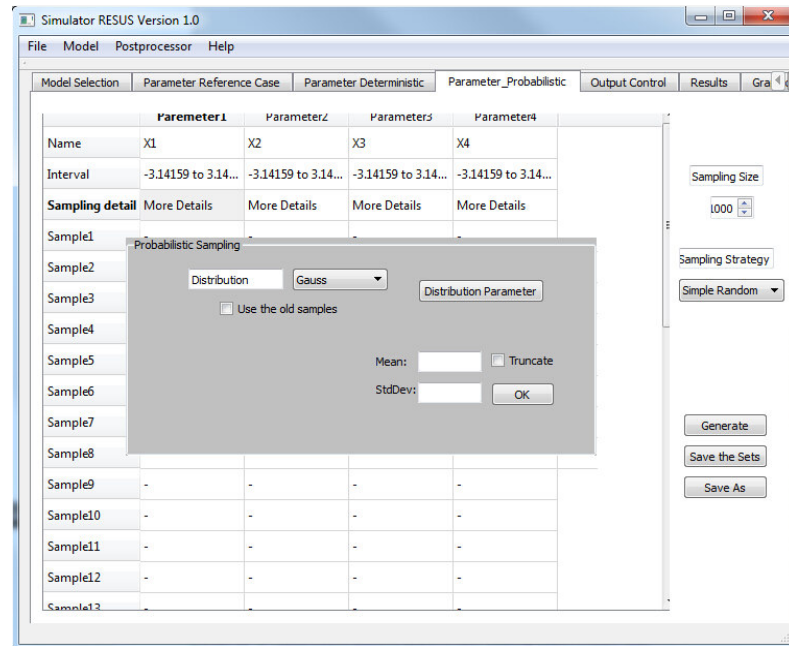


Abb. 4.8b Fenster für die Einstellung der Verteilungsfunktion eines probabilistischen Parameters bei der Wahl der Verteilung „Gauss“ oder „log\_Gauss“

Bei der Wahl der Verteilungen „uniform“ und „log-uniform“ sind „Min“- und „Max“ als Untergrenze und Obergrenze der Verteilung anzugeben. Bei der Wahl von „Gauss“ muss der Nutzer den Mittelwert und die Standardabweichung der Verteilung eingeben. Bei „log\_Gauss“ ist der Mittelwert und die Standardabweichung zur originalen Gaußverteilung zu geben, RESUS transformiert anschließend die Samples mit den Angaben zur logarithmischen Gaußverteilung.

Neben den Gleich- und Gaußverteilungen ist zuletzt noch die Dreiecksverteilung wählbar. Hier werden drei Verteilungsmaße gebraucht: (siehe Abb. 4.9) Minimaler und maximaler Wert sowie der Modalwert.

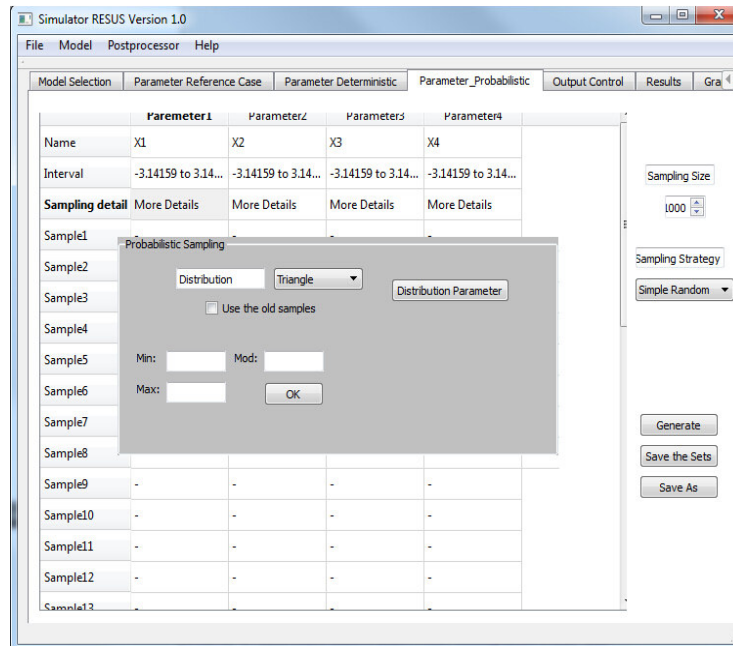


Abb. 4.9 Fenster zur Eingabe der Werte für die Dreieckverteilung

Die eingegebenen Verteilungen und die eingetragenen Attribute werden von RESUS auf die Einhaltung der vorher festgelegten Wertintervalle der Parameter kontrolliert. Liegen die eingestellten Verteilungsmaße außerhalb des Wertintervalls, so erfolgt eine Abfrage, ob dies vom Nutzer so gewollt ist. Ein Dialogfenster wird eingeblendet (siehe Abb. 4.10). Wenn der Nutzer bei dem Fenster „yes“ wählt, werden die Samples trotz des Intervallkonflikts nach den eingegebenen Werten generiert. Wenn „No“ gewählt wird, leitet RESUS den Benutzer zurück zum Eingabeprozess der Verteilungseigenschaften.

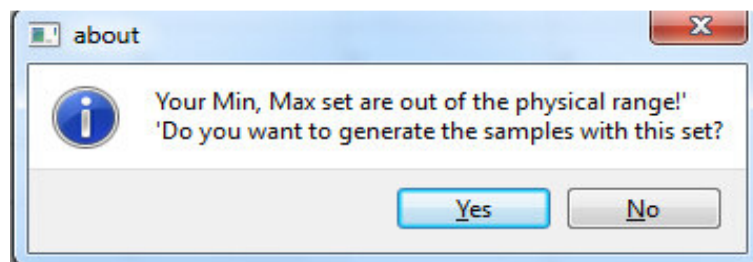


Abb. 4.10 Fenster bei Verletzung des festgelegten Wertintervalls

Bei fehlerhaften Eingaben, z.B. Minimalwert größer als Maximalwert wird eine Warnung eingeblendet (siehe Abb. 4.11).



Abb. 4.11 Warnung bei Inkonsistenz der hinterlegten Werte (Minimal-, Maximal- oder Modalwert)

Bei der Gaußverteilung- und der logarithmischen Gaußverteilung müssen Mittelwert und Standardabweichung angegeben werden (siehe Abb. 4.12). Im dafür vorgesehenen Fenster gibt es überdies die Option „Truncate“ (Abschneiden) zur Einstellung der Wertebereiche der Zufallsparameter. Ist sie nicht angekreuzt, nehmen die Verteilungsfunktionen uneingeschränkt Werte im Intervall  $(-\infty, +\infty)$  an. Ist „Truncate“ angekreuzt, so werden nur Zufallswerte innerhalb der vorher festgelegten physikalischen Wertebereiche akzeptiert. Alle probabilistischen Parameter haben wie oben erläutert ein physikalisches Wertintervall, das bei der Auswahl von „truncate“ zugrundegelegt wird.

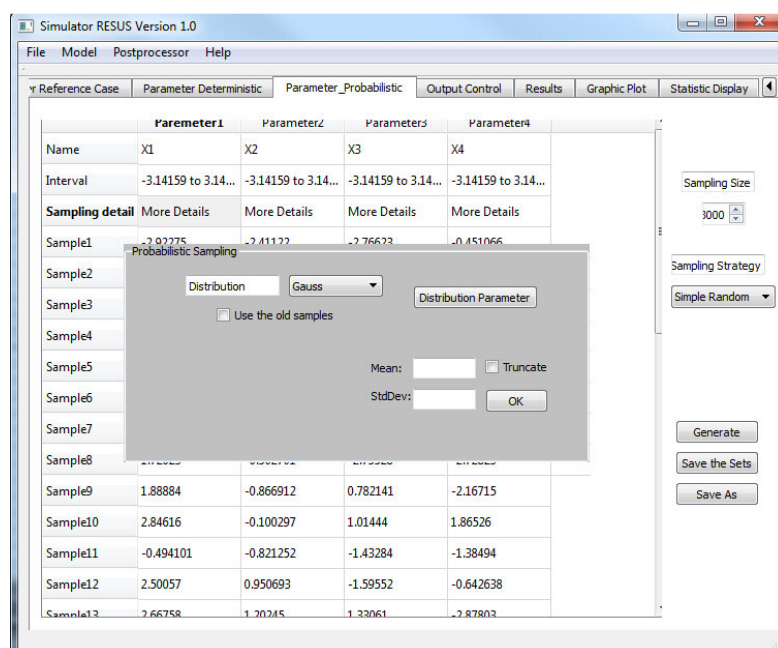


Abb. 4.12 Fenster für die Einstellung der statistischen Verteilung eines probabilistischen Parameters

Die Option „Use the old samples“ kann genutzt werden, wenn ein RESUS-Modell aus einer Datei wiederhergestellt wird. Ist „Use the old samples“ für einen Parameter gewählt, so werden dessen in der Datei gespeicherte frühere Zufallswerte wieder zur Simulation verwendet. Ein Beispiel für die Nutzung dieser Option wird in Kapitel 4.3 vorgestellt.

Im Fenster für die Einstellung der statistischen Verteilung eines probabilistischen Parameters werden die Eingaben, ähnlich wie oben beschrieben, wieder auf Konsistenz geprüft. Bei aktivierter „Truncate“-Option sollte der Mittelwert normalerweise im Wertebereich des physikalischen Intervalls liegen. Ist das nicht der Fall, so erscheint eine Warnung (siehe Abb. 4.13). Ignoriert man diese („yes“ anzuklicken), gerät das Programm möglicherweise in eine Endlosschleife bei der Werteerzeugung. Es liegt daran, dass es durch die Inkonsistenz des Mittelwertes möglich ist, dass entsprechend der Verteilung alle Zufallswerte außerhalb der eingestellten Intervallgrenzen erzeugt werden. „Truncate“ prüft aber die Grenzbedingung und verwirft die Werte. Die Werteerzeugung kann in der Folge nicht abgeschlossen werden.

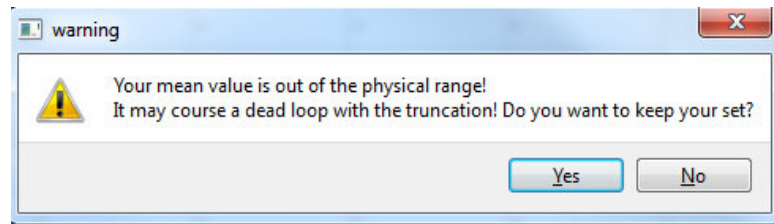


Abb. 4.13 Anzeige einer Warnung, wenn der festgelegte Mittelwert der Gaußverteilung außerhalb des vorgegebenen Wertebereichs liegt

Sind die Einstellungen für alle probabilistischen Parameter vollständig, kann der Benutzer durch Drücken auf „Generate“ Zufallswerte erzeugen lassen. Bei unvollständigen Parameter-Eigenschaften (z.B. unvollständig festgelegten Verteilungen) ruft RESUS ein Warnfenster auf und kehrt zum Einstellungs-Tab zurück. Nach der Erzeugung der Zufallswerte werden diese in einer Tabelle zur manuellen Kontrolle aufgelistet. Wenn die Anzahl der Zufallswerte 30 übersteigt, werden nur die ersten 30 Werte aufgelistet.

Der Knopf „Save“ und „Save As“ sind für die Speicherung der Zufallswerte und des Modelle. Diese Funktion wird in Kapitel 4.6 beschrieben.

Im nächsten Schritt müssen die Einstellungen für die Darstellung der Ausgabeparameter gewählt werden. Die Ausgabeparameter werden im nächsten Tab in der Tabelle „Output Control“ aufgelistet (siehe Abb. 4.14).



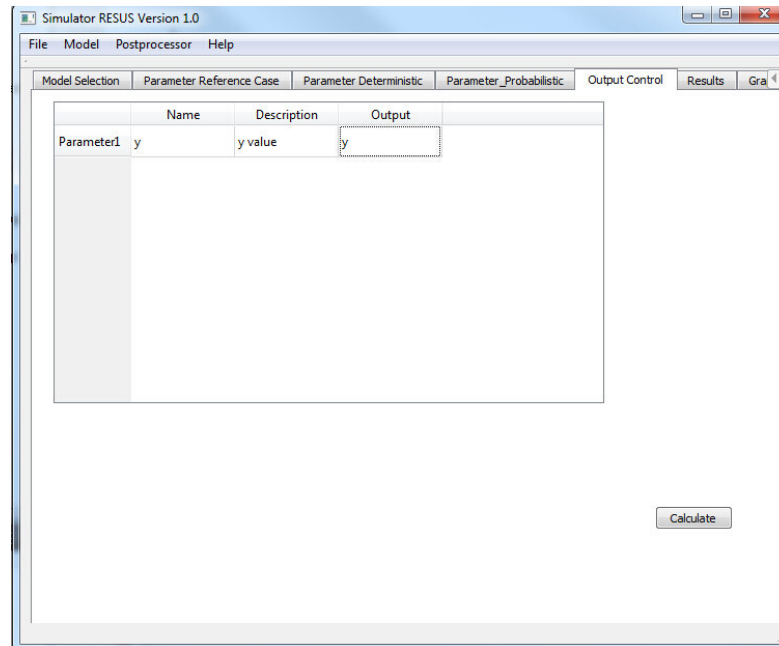


Abb. 4.14 Tab für die Einstellung für die Darstellung der Ausgabeparameter der Simulation

In der „Output Control“-Tabelle werden die Namen, die Beschreibungen und die Ausgabeeinstellungen für alle Ausgabeparameter aufgelistet. Das Ishigami-Beispielmodell hat nur die Ausgabegröße „y“ (siehe Gleichung 4-1), deswegen ist hier nur „y“ als Ausgabeparameter dargestellt. Der Benutzer muss in der „Output“ Zelle „y“ eingeben. „y“ („yes“) an dieser Stelle bedeutet, dass der entsprechende Ausgabeparameter weitergereicht wird. Der Parameter kann somit beim späteren Post-Processing dargestellt und analysiert werden. „n“ oder andere Buchstaben dagegen bedeuten, dass der Nutzer den entsprechenden Parameter für die Analyse nicht braucht. Er wird folglich nicht berechnet und bindet keine Rechenkapazitäten. Der Default-Wert an dieser Stelle ist eine „n“ für „nein“.

Nach der Einstellung der Ausgabeparameter ist das Pre-Processing abgeschlossen. Mit einem Druck auf den „Calculate“ Knopf wird die Simulation ausgeführt. RESUS berechnet den ersten Durchlauf stets mit den Referenzwerten der probabilistischen Parameter. So wird die vom Nutzer genau definierte Berechnung zuerst ausgeführt, um bei später eventuell auftretenden Fehlern die Modelleinstellung kontrollieren zu können. Bei unserem Beispiel wird das Ishigami-Modell mit dem Referenzwert und den 3000 gegebenen Samplewerten berechnet, es wird also 3001-mal y ausgegeben.

Nach dem Abschluss der Berechnungen beginnt die dritte Phase: das Post-Processing.

Der Tab mit „Graphic Plot“ beinhaltet die Darstellung der zeitabhängigen Ergebnisse. Da das Ergebnis aus der Berechnung des Ishigami-Modells nicht zeitabhängig ist, wird der Tab in diesem Beispiel nicht gebraucht. Die Inhalte dieses Tabs werden weiter anhand anderer Beispiele vorgestellt. In unserem Ishigami-Beispielmodell gehen wir direkt zum „Statistic Display“-Tab über.

Im „Statistic Display“-Tab ist die statistische Analyse und die Sensitivitätsanalyse zu finden. Der Benutzer kann hier mit den Rechenergebnissen eine Histogramm- und Streudiagramm-Analyse durchführen lassen. Die Abb. 4.15 zeigt die Ansicht des „Statistic Display“-Tabs. Der Großteil der Fläche wird durch den Darstellungsbereich eingenommen. Rechts oben ist ein Textfeld mit Informationen über die aktuelle Darstellung. Unter dem Textfeld befindet sich ein Auswahlménü. Darin wird die Verarbeitungsart der Rechenergebnisse festgelegt, die vom Wesen der darzustellenden Daten abhängt.

„Quantity of interest“ wird hier als die Aufbereitungsmethode der zeitabhängigen Ergebnisse genannt. Es kann zwischen „Max per realization“, „Integral“ gewählt werden. Da unser Ishigami-Beispielmodell keine zeitabhängigen Ergebnisse hat (pro Rechendurchlauf gibt es nur ein skalaras Ergebnis), wird die „Quantity of interest“ auf „Max per realization“ gesetzt. Diese Berechnungsmethode der „Quantity of interest“ errechnet den „maximalen“ Wert von „y“, was in diesem Beispiel genau der y-Wert ist.

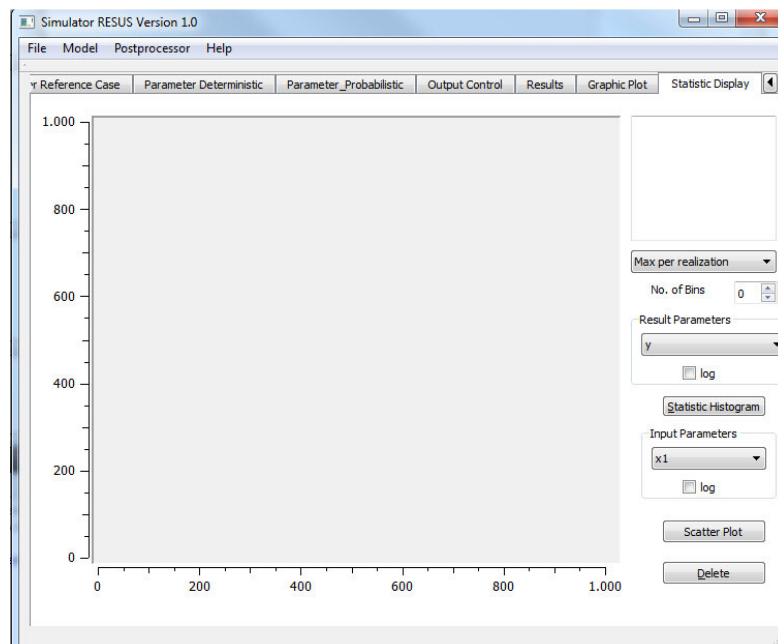


Abb. 4.15 Tab „Statistic Display“

Rechts in Abb. 4.15: Das Auswahlménü für die Art der Aufbereitung der Rechenergebnisse zum Zweck der statistischen Darstellung. Weiterhin ist im gleichen „Statistic Display“-Tab rechts, unterhalb des beschriebenen Auswahlménüs, eine Eingabefläche, die die Darstellungsart „Histogramm“ betrifft und die Anzahl der Wertklassen festlegt („No. of Bins“). Mit einem Histogramm kann ermittelt werden, wie die Rechenergebnisse verteilt sind. Mit einem Klick auf den Knopf „Delete“ wird die grafische Darstellung gelöscht. Dies ist erforderlich, um eine neue Grafik erstellen zu können.

Bei unserem Beispiel mit 3001 Rechenergebnissen wählen wir zunächst 40 Wertklassen und erzeugen die Histogramm-Darstellung durch Anklicken des „Statistic Histogram“-Knopfs. Das Ergebnis unter Berücksichtigung der „Max per realization“-Einstellung sieht man in Abb. 4.16.

Im beschriebenen Textfeld werden die statistischen Quantile dargestellt. Die Berechnung der statistischen Quantile durch RESUS wird in Kapitel 5 vorgestellt.

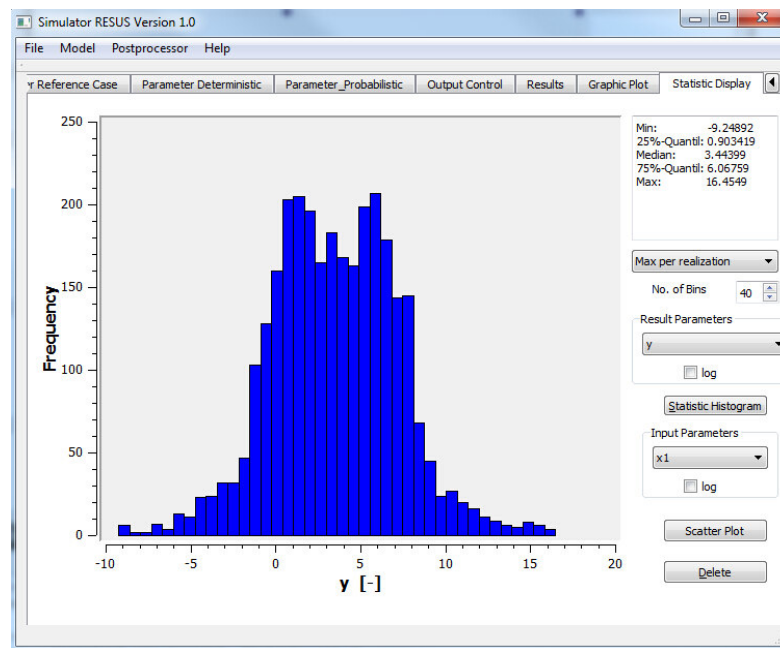


Abb. 4.16 Histogramm aus 40 Wertklassen mit der „Quantity of interest“ „Max per realization“

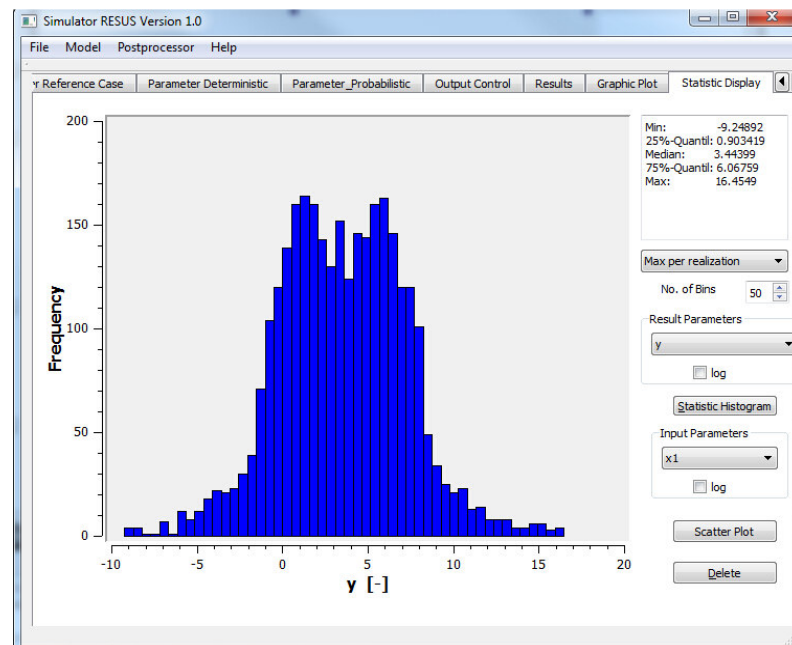


Abb. 4.17 Histogramm aus 50 Wertklassen mit der „Quantity of interest“ „Max per realization“

Wir wollen in unserem Beispiel nur zu Demonstrationszwecken noch ein Histogramm mit 50 Werteklassen erzeugen (siehe Abb. 4.17).

Wie zu sehen weist das 50-Klassen-Histogramm inhaltlich keinen großen Unterschied zum gezeigten 40-Klassen-Histogramm auf. Die Rechenergebnisse offerieren eine mögliche Bimodale-Verteilung, jedoch kann aus einer visuellen Auswertung des Histogramms kein eindeutiges Urteil gefällt werden.

Neben der Histogrammdarstellung der Rechenergebnisse ist auch eine Darstellung als Streudiagramm (Scatterplot) möglich. Die Rechenergebnisse werden dabei auf der Ordinate über einem der Eingangsparameter (in diesem Beispiel:  $X_1$ ,  $X_2$ ,  $X_3$  oder  $X_4$ ) auf der Abszisse aufgetragen. Der Eingangsparameter kann im Drop-Down-Menü „Input Parameter“ festgelegt werden (siehe Abb. 4.18).

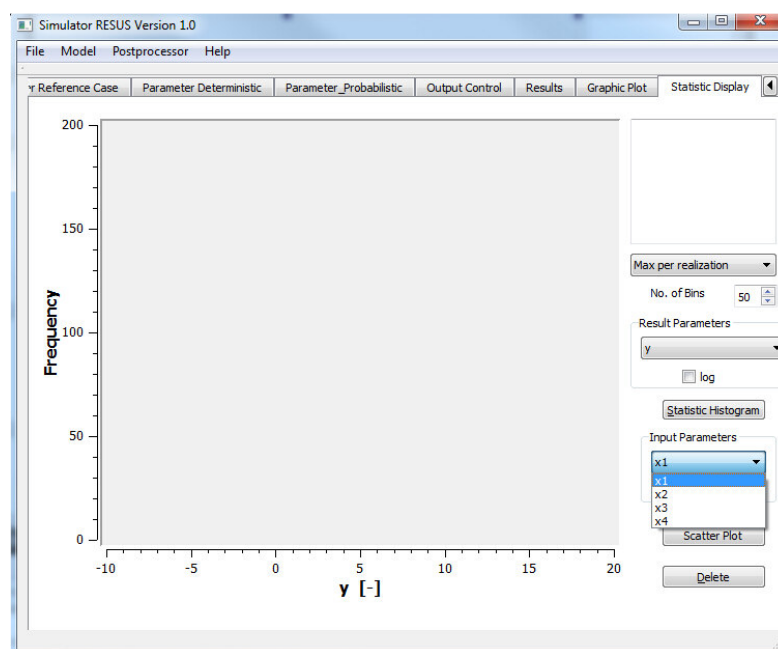


Abb. 4.18 Rechts, unten: Auswahl des Eingangsparameters zum Streudiagramm (Scatterplot). Alle probabilistischen Parameter stehen zur Auswahl

Nach der Wahl eines Eingangsparameters erzeugt ein Klick auf den Knopf „Scatterplot“ das Streudiagramm aus den Zufallswerten des gewählten Eingangsparameters und den Rechenergebnissen. Die Sensitivitätsanalyse ist eine der Hauptaufgaben von RESUS. Sie liefert allgemein Aussagen dazu, wie stark die Eingangsbedingungen die Rechenergebnisse beeinflussen, also wie sensitiv, sensibel bzw. empfindlich ein System reagiert.

In unserem Ishigami-Modell gibt es 4 Eingangsparameter:  $X_1$ ,  $X_2$ ,  $X_3$  und  $X_4$  (mit  $X_4$  als Dummy-Parameter). Mit jedem wird jeweils ein Streudiagramm erstellt (siehe Abb. 4.19 a, b, c und d). Auf der Ordinate werden die aufbereiteten Rechenergebnisse und auf der Abszisse die Eingangsparameter aufgetragen.

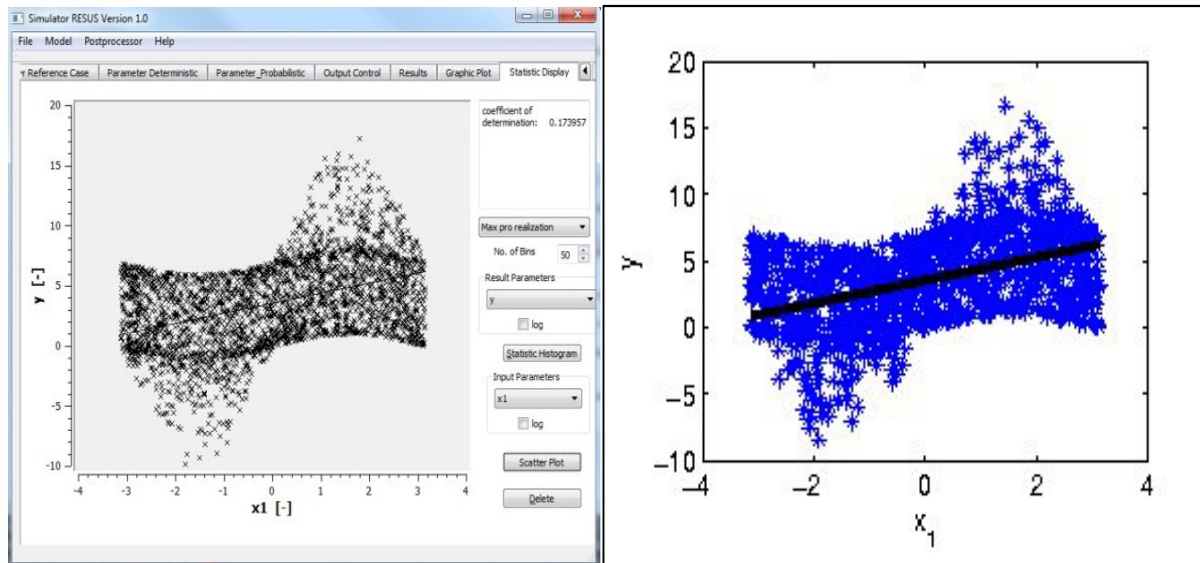


Abb. 4.19a Streudiagramm zum Ishigami-Modell,  $X_1$  gegen  $Y$ , links ist die Darstellung in RESUS, Rechts ist die Darstellung der Ergebnisse aus der Literatur [22]

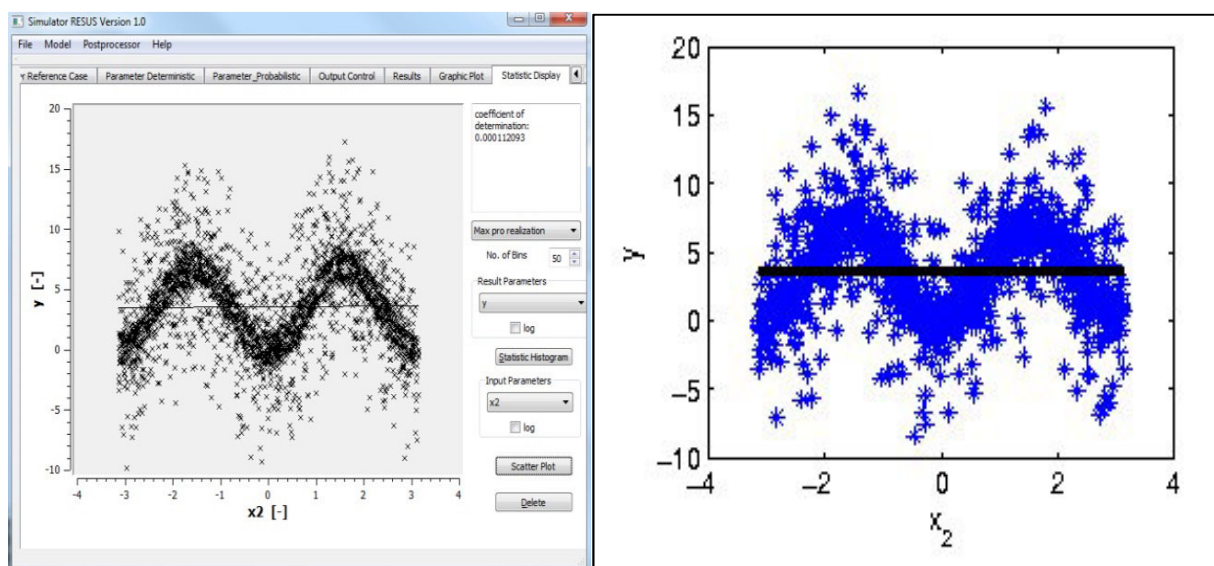


Abb. 4-19b Streudiagramm zum Ishigami-Modell,  $X_2$  gegen  $Y$ , links ist die Darstellung in RESUS, Rechts ist die Darstellung der Ergebnisse aus der Literatur [22]



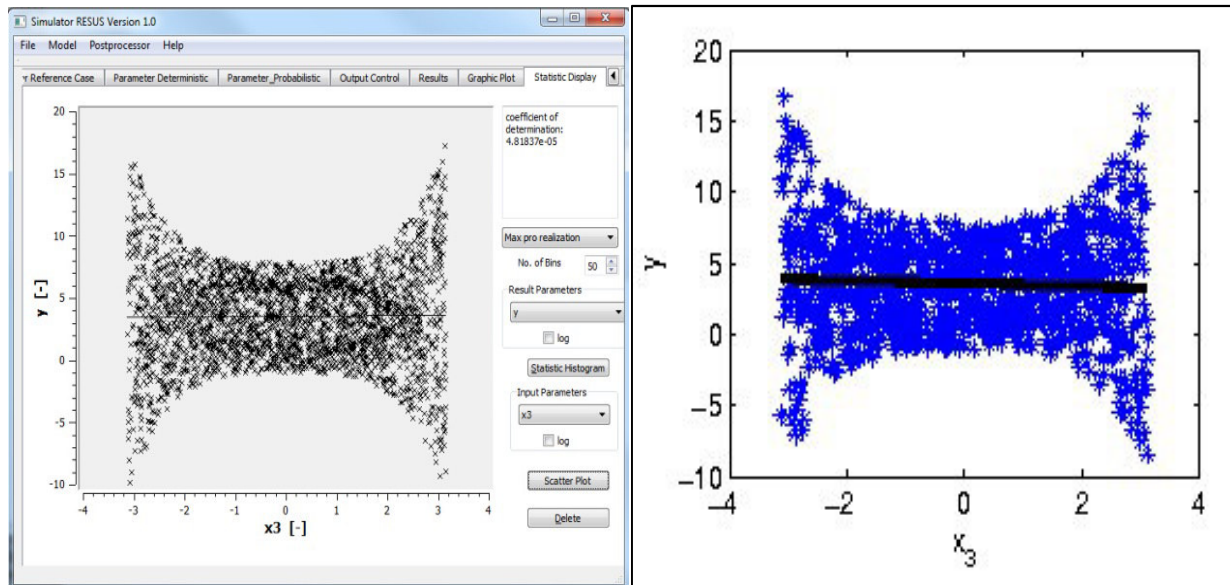


Abb. 4-19c Streudiagramm zum Ishigami-Modell,  $X_3$  gegen  $Y$ , links ist die Darstellung in RESUS, Rechts ist die Darstellung der Ergebnisse aus der Literatur [22]

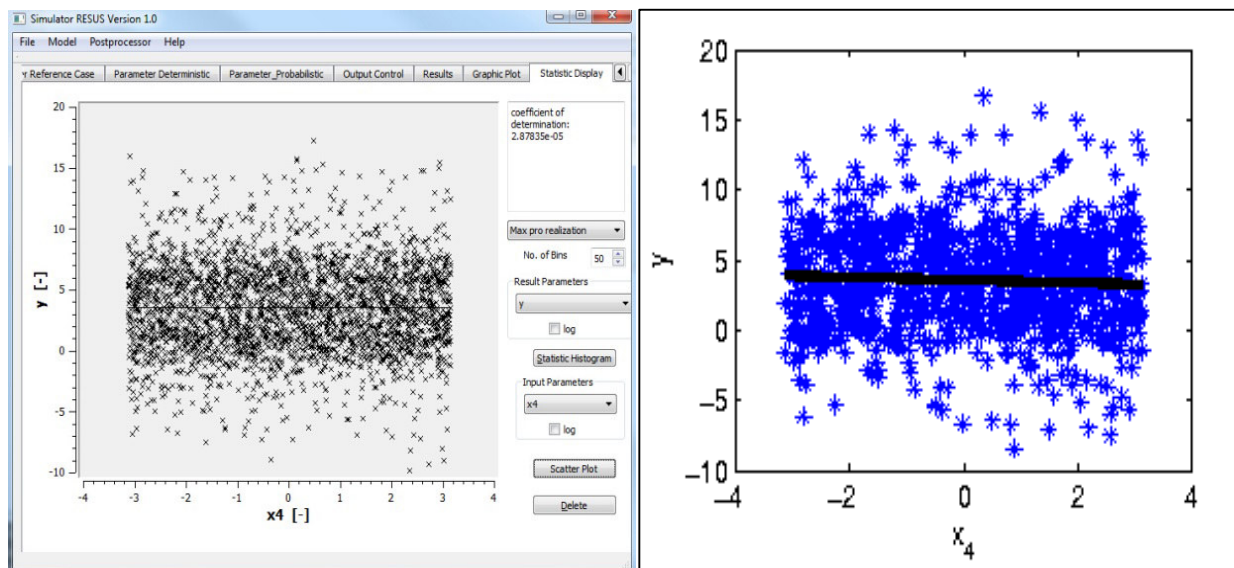


Abb. 4-19d Streudiagramm zum Ishigami-Modell,  $X_4$  gegen  $Y$ , links ist die Darstellung in RESUS, Rechts ist die Darstellung der Ergebnisse aus der Literatur [22]

Auf den vergleichenden Abb. 4.19a, b, c und d ist zu erkennen, dass für alle Eingangsparameter die Streudiagramme von RESUS der Literatur ähneln und einen gleichen Trendverlauf aufweisen [22].

Der von RESUS angezeigte Trendverlauf, der als dünner schwarzer Strich durch die Punktwolke läuft, ist eine lineare Regressionsgerade. Zusätzlich wird der Bestimmtheitsmaß  $R^2$  im Textfeld rechts oben angezeigt.

Dies sind die Bestimmtheitsmaße zwischen Ein- und Ausgabeparametern in unserem Modell:

$R^2 \text{ } X_1\text{-}Y$	0,1739
$R^2 \text{ } X_2\text{-}Y$	0,0001
$R^2 \text{ } X_3\text{-}Y$	$4,81837 \cdot 10^{-5}$
$R^2 \text{ } X_4\text{-}Y$	$2,87835 \cdot 10^{-5}$

Man erkennt anhand der nahe an 0 liegenden Werte, dass bei Parametern 2 bis 4 kein linearer Zusammenhang zwischen den Eingangsgrößen und den Rechenergebnissen besteht.

Die letzte zu beschreibende ist die „log“- Auswahl im „Result Parameters“-Bereich. Bei gesetztem Haken wird die gewählte Ausgabegröße für die Darstellung logarithmisch zur Basis 10 transformiert. In unserem Ishigami-Beispiel wird eine logarithmische Transformation nicht gebraucht. Sinn und Zweck der logarithmischen Transformation wird daher anhand späterer Beispiele gezeigt.

In der weiterzuentwickelnden Version von RESUS werden weitere Sensitivitätsmaße implementiert.

## 4.2 Level-E in RESUS

Level-E ist ein Programmcode für die Sicherheitsanalyse von. Er wird als „Probabilistic System Assessment Code“ (PSAC) bezeichnet. Der Programmcode wurde von Nuclear Energy Agency (NEA) im Jahr 1989 entwickelt und simuliert die Schadstoff-Freisetzung und den Transport in einem stark vereinfachten virtuellen Endlagersystem. Das darin berechnete Endlagersystem-Modell ist stark vereinfacht. Es bildet kein reales Endlagersystem nach, sondern dient nur zum Test für probabilistische Analysemethoden [24].

Das Modell enthält eine Schadstoffquelle (Quellterm), ein eindimensionales Transportmodell und ein Biosphärenmodell mit einer Trinkwasserquelle. In RESUS ist das Level-E-Modell fest implementiert und kann über die Benutzeroberfläche leicht bearbeitet werden. Die GUI greift dabei im vollen Umfang auf die Funktionen von RESUS zurück. Dank seiner geringen Komplexität ist das Level-E Modell in RESUS für verschiedene Aufgabebereiche vorgesehen:

1. Es soll als Beispielmmodell dienen, an dem die Transformation zeitabhängiger Daten getestet werden kann.
2. Es soll allgemein als leicht verständliches und schnell aufzurufendes Beispielmmodell bei Lehre und Ausbildung verwendet werden.

3. Das modulare Level-E Modell soll auch bei der Bearbeitung anderer Analysemodelle in RESUS als Hilfe dienen, da seine Komponenten gut in andere Modelle implementiert werden können.

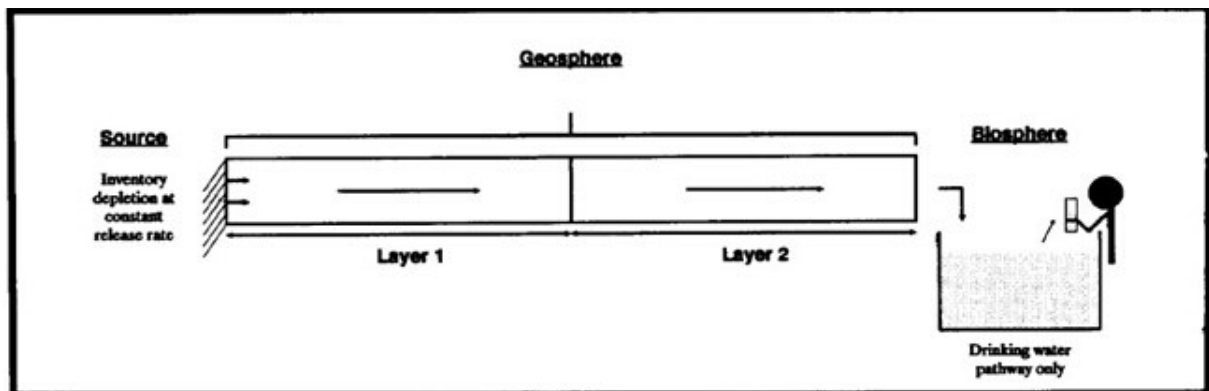


Abb. 4.20 Aufbau eines Level-E Modells eines stark vereinfachten Endlagersystems [24]

Von einem Quellterm links im Bild werden Radionuklide freigesetzt und durch eine Grundwasserströmung nach rechts durch zwei geologische Schichten (Layer1, Layer2) transportiert. Dadurch wird schließlich das Trinkwasser in der Biosphäre (Biosphere) kontaminiert [24].

Dieses Modell besteht aus drei Teilen: dem Quellterm (Nahfeld), dem eindimensionalen Transport in zwei Grundwasserleitern (Fernfeld) und dem Biosphärenmodell.

Das Level E-Modell betrachtet stets vier repräsentative Radionuklide. Zum einen die  $\alpha$ -Zerfallskette Np-237, U-233 Th-229 und zum anderen I-129.

Gleichung 4-2 gibt die Freisetzung aus dem Quellterm der beiden Nuklide: Np-237 und I-129 entlang des Freisetzungspfads an.

$$\begin{aligned} \frac{\partial c_i}{\partial x} &= -\lambda_i \times c_i, & t \leq T \\ \frac{\partial c_i}{\partial x} &= -\lambda_i c_i - k_i c_i, & t > T \end{aligned} \quad (4-2)$$

Hier ist  $c_i$  [mol/m<sup>3</sup>] die Konzentration des Radionuklids i im Quellterm.  $\lambda_i$  [Jahr<sup>-1</sup>] ist der Zerfallskoeffizient des jeweiligen Nuklids.  $T$  [Jahr] ist der Zeitpunkt, zu dem der Abfallbehälter versagt. Man sieht anhand der Gleichung, dass die Konzentration des Radionuklids vor dem Behälterversagen ( $t \leq T$ ) lediglich durch den Zerfallsprozess abnimmt, nach dem Behälterversagen ( $t > T$ ) aber zusätzlich durch das  $-k_i c_i$  Term fällt. Das liegt an der beginnenden Freisetzung in die Geosphäre,  $k_i$  [a<sup>-1</sup>] bezeichnet hier die Freisetzungsrate.

Gleichung 4-3 gibt die Freisetzung aus dem Quellterm der beiden Tochternuklide U-233 und Th-229 entlang des Freisetzungspfads an:



$$\frac{\partial c_i}{\partial x} = \lambda_{i-1} * c_{i-1} - \lambda_i * c_i, \quad t \leq T$$

$$\frac{\partial c_i}{\partial x} = \lambda_{i-1} * c_{i-1} - \lambda_i * c_i - k_i c_i, \quad t > T \quad (4-3)$$

$\lambda_{i-1} [a^{-1}]$  ist der Zerfallskoeffizient des Mutternuklids und  $c_{i-1} [mol/m^3]$  ist die Konzentration des Mutternuklids. Die anderen Terme sind identisch wie in Gleichung 4-3.

Bei der Mengenbetrachtung der in Gleichung 4-2 beschriebenen Nuklide gibt es lediglich eine Abnahme durch die Zerfallsprozesse. Dies ist anders in Gleichung 4-3, die für die Tochternuklide U-233 und Th-229 gilt. Hier werden zusätzlich zum Zerfalleffekt auch Zuwachseffekte berücksichtigt.

Im Quellterm von Level E werden sowohl der Anfangszeitpunkt der Freisetzung als auch die Freisetzungsrates  $k_i$  als probabilistische Parameter behandelt. Die anderen Parameter im Quellterm, wie z.B. die Anfangskonzentrationen, sind deterministisch und gehen mit konstanten Werten ein.

Ein Teil der Radionuklide wird nach Durchdringung der geologischen Schichten die Biosphäre erreichen. Die Transportmodellierung für die Geosphäre liefert hier ein eindimensionales Transportmodell, das die Advektion, Dispersion und die Retardationsprozesse berücksichtigt. Die mathematische Darstellung des Modells ist wie folgt:

$$\frac{\partial c_i^k}{\partial t} R_i^k + \frac{\partial c_i^k}{\partial x} v^k - \frac{\partial^2 c_i^k}{\partial x^2} d^k v^k = -\lambda_i R_i^k C_i^k \quad (4-4)$$

$$\frac{\partial c_i^k}{\partial t} R_i^k + \frac{\partial c_i^k}{\partial x} v^k - \frac{\partial^2 c_i^k}{\partial x^2} d^k v^k = -\lambda_i R_i^k C_i^k + \lambda_{i-1} R_{i-1}^k C_{i-1}^k \quad (4-5)$$

Die Gleichung 4-4 beschreibt den Transport der Mutternuklide (Np-237, I-129) und ihre Konzentrationsverteilung in der Geosphäre.  $R_i^k [-]$  ist der Retardationsfaktor für das jeweilige Radionuklid in der jeweiligen Geosphärenschicht und modelliert folglich den Sorptionsprozess (lineares Gleichgewicht, Henry-Isotherme).  $v^k [m/a]$  steht für die Wasserfließgeschwindigkeit in der  $k$ -ten geologischen Schicht. Der gesamte Term  $\frac{\partial c_i^k}{\partial x} v^k$  beschreibt den advektiven Transport der Schadstoffe.  $d^k$  ist die Dispersionslänge der Geosphäre-Schicht, der gesamte Term  $\frac{\partial^2 c_i^k}{\partial x^2} d^k v^k$  beschreibt den Dispersionsprozess.  $\lambda_i R_i^k C_i^k$  ist der Term für den Schadstoffabbau durch den Zerfall des Radionuklids.

Jod hat in diesem Modell einen zufälligen Retardationsfaktor:  $R_I^k$ . Für den Retardationsprozess der Np-Kette werden die Basis-Retardationsfaktoren  $R_{0i}^k$  für alle drei Nuklide zuerst als feste Werte gegeben. Diese wird zudem mit einem probabilistischen Wert  $\gamma$  als Umrechnungsfaktor für alle drei Nuklide multipliziert. Der Retardationsfaktor für die Simulation ergibt sich dann aus der Gleichung:  $R_i^k = \gamma * R_{0i}^k$ .

In der Gleichung 4-5 wird die Konzentration der Tochternuklide (U-233 und Th-229) berechnet. Im Teilmodell das die Geosphäre betrifft sind die Wasserfließrate  $v^k$ , die Schichtlänge  $l^k$  und der Retardationsfaktor  $R_i^k$  probabilistisch. Die anderen Parameter im Modell werden deterministisch berechnet.

Wenn die Radionuklide zum Rand der Geoschicht transportiert werden, gelangen sie in die Biosphäre. Das von Level-E verwendete stark vereinfachte Biosphärenmodell nimmt an, dass Schadstoffe, die einmal an die Erdoberfläche gelangt sind, dort einen kleinen Fluss kontaminieren. Das Flusswasser wird von Menschen als Trinkwasser verwendet und mit ihm eine individuelle Schadstoffdosis inkorporiert.

Mit Gleichung 4-6 wird die individuelle Dosis in der Biosphäre berechnet:

$$Dose(t) = \frac{P}{w} \beta_i C_i(t) \quad (4-6)$$

$w$  ist hier die Wassermenge [ $m^3/Jahr$ ] des betrachteten Flusses;  $\beta_i$  ist der jeweilige Dosisfaktor der Radionuklide,  $p$  ist die von einem Menschen im Jahr getrunkene Wassermenge [ $m^3$ ].  $C_i(t)$  ist die zeitabhängige Konzentration des betrachteten Nuklids im Flusswasser [ $mol/m^3$ ].

In diesem Biosphärenmodell ist die betrachtete Wassermenge der einzige probabilistische Parameter. Für die anderen Parameter werden im Level-E-Modell feste Werte angenommen.

Die Gesamtdosis aus den vier betrachteten Radionukliden ergibt sich wie schon in Gleichung 4-6 aus der Summation der einzelnen Dosiswerte:

$$Dose(total) = \sum_{i=1}^4 Dose_i \quad (4-7)$$

$i$  ist hier der Radionuklidindex (*I-129, Np-237, U-233, Th-229*).

Das Level-E Modell bietet eine analytische Lösung zur Verfügung.

Die Erstellung und Analyse des Level-E Modells in RESUS soll in diesem Kapitel schrittweise vorgestellt werden.

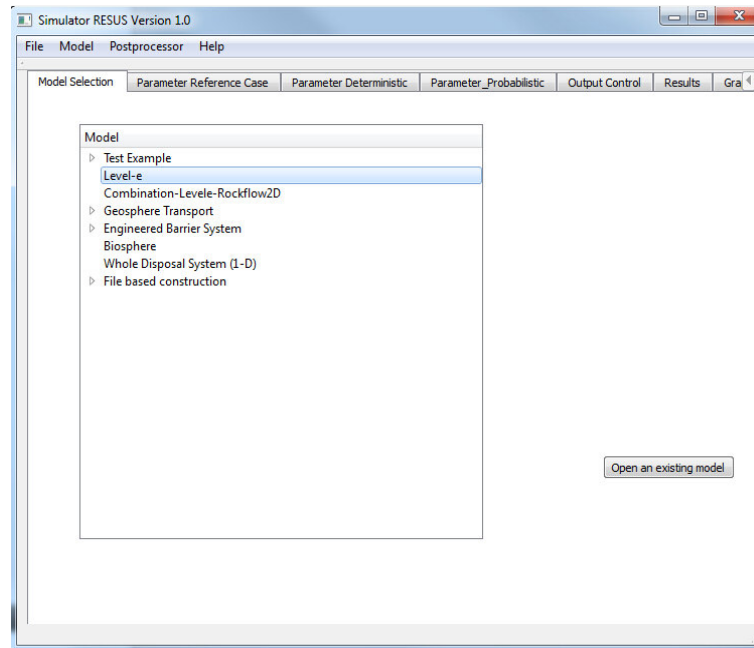


Abb. 4.21 Auswahl des Level-E Modells im „Model Selection“-Tab

Ähnlich wie beim Ishigami-Modell sucht der Benutzer das Level-E Modell zunächst aus der Baumstruktur im „Model Selection“-Tab aus und initialisiert es dann mit einem Doppelklick auf „Level-E“. Daraufhin springt RESUS zum zweiten Tab mit der Bezeichnung „Parameter Reference Case“. Aus den oben erwähnten Modellgleichungen ergeben sich insgesamt zwölf probabilistische Parameter, die in der nun sichtbaren Tabelle bereits eingetragen sind (siehe Abb. 4.22):

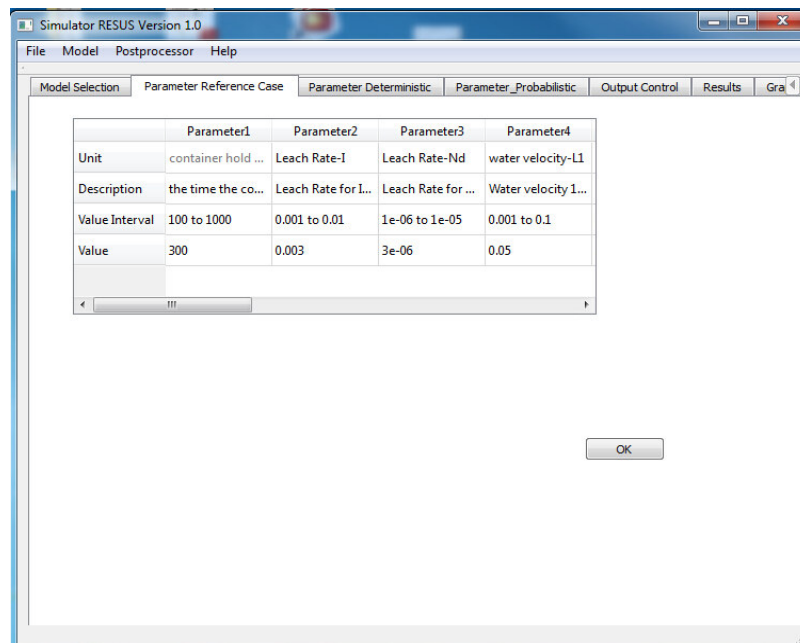


Abb. 4.22 Tabelle zur Einstellung der deterministischen Werte der probabilistischen Parameter

In dieser Tabelle müssen die Referenzwerte für die zwölf Parameter eingegeben werden. Die Werte für unser Beispielmmodell sind in der folgenden Tabelle aufgeführt.

Tabelle 4-1 Referenzwerte für die probabilistischen Parameter im Level-E Modell

Parameter	Beschreibung	Einheit	Referenzwert
$T$	Zeitdauer bis zum Behälterversagen	Jahr	300
$K_I$	Freisetzungsrates von Jod	l/a	0,003
$K_c$	Freisetzungsrates der Np-Kette	l/a	$3 \cdot 10^{-6}$
$v^1$	Fließgeschwindigkeit in der Geoschicht 1	m/a	0,05
$l^1$	Länge der Geoschicht 1	m	200
$R_I^1$	Retardationsfaktor für Jod in Geoschicht 1	-	3
$r_c^1$	Retardationsfaktor für die Np-Kette in Geoschicht 1	-	5
$v^2$	Fließgeschwindigkeit in Geoschicht 2	m/a	0,03
$l^2$	Länge der Geoschicht 2	m	100
$R_I^2$	Retardationsfaktor für Jod in der Geoschicht 2	-	3
$r_c^2$	Allgemeiner Umrechnungsfaktor für den Retardationsfaktor der Np-Kette in Geoschicht 2	-	10
W	Betrachtete Volumen von Flusswasser in der Biosphäre pro Jahr	$m^3/a$	$3 \cdot 10^6$

Es werden von RESUS nur die probabilistischen Parameter im Level-E Modell konfiguriert. Die deterministischen Parameter sind bei Level-E nicht editierbar. Aus diesem Grund wird der Tab „Parameter\_Deterministic“ wie im Ishigami-Modell nicht gebraucht.

Wie in unserem vorherigen Ishigami-Beispielmmodell dient das Fenster „Parameter\_Probabilistic“ auch zur Einstellung der statistischen Eigenschaften der Parameter. Die statistischen Verteilungen und Parameter werden wie zuvor einzeln festgelegt (siehe Abb. 4.23). Die in diesem Beispiel eingegebenen Werte sind in der folgenden Tabelle aufgeführt.

Tabelle 4-2 Probabilistischen Parameter der Level-E Modellkomponenten

Parameter	Beschreibung	Einheit	Verteilung	Wertintervall
$T$	Zeitdauer bis zum Behälterversagen	Jahr	Gleich	100-1000
$K_I$	Freisetzungsrates von Jod	l/a	Log-gleich	$0,001-10^{-2}$
$K_C$	Freisetzungsrates der Np-Kette	l/a	Log-gleich	$10^{-6}-10^{-5}$
$v^1$	Abstandfließgeschwindigkeit in Geoschicht 1	m/a	Log-gleich	0,001-0,1
$l^1$	Länge der Geoschicht 1	m	Gleich	100-500
$R_I^1$	Retardationsfaktor für Jod in Geoschicht 1	-	Gleich	1-5
$r_C^1$	Retardationsfaktor der Np-Kette in Geoschicht 1	-	Gleich	3-30
$v^2$	Abstandfließgeschwindigkeit in Geoschicht 2	m/a	Log-gleich	0,01-0,1
$l^2$	Länge der Geoschicht 2	m	Gleich	50-200
$R_I^2$	Retardationsfaktor für Jod in Geoschicht 2	-	Gleich	1-5
$r_C^2$	Allgemeiner Umrechnungsfaktor für den Retardationsfaktor der Np-Kette in Geoschicht 2	-	Gleich	3-30
$W$	Betrachtete Volumen von Flusswasser in der Biosphäre pro Jahr	m <sup>3</sup> /a	Log-gleich	$10^5-10^7$

Die probabilistischen Parameter von Level-E sind alle gleichverteilt oder logarithmisch gleichverteilt. Nach der Festlegung der jeweiligen Verteilungen werden die Zufallswerte mit einem Klick auf den Knopf „Generate“ erzeugt (Siehe Abb. 4.23). Die Anzahl der Zufallswerte in diesem Beispiel wird auf 300 festgelegt.

Name	Interval	Sampling detail	Parameter1	Parameter2	Parameter3	Parameter4	Parameter5
Sample1	520.673	0.00352535	7.91581e-06	0.00170098	311.298		
Sample2	552	0.00878617	1.68408e-06	0.00418842	329.296		
Sample3	104.102	0.00246711	1.73392e-06	0.00246158	255.311		
Sample4	793.146	0.00369062	1.5305e-06	0.00635746	487.214		
Sample5	446.466	0.00333577	1.64009e-06	0.00362701	296.527		
Sample6	610.369	0.00403536	2.6154e-06	0.00969933	275.641		
Sample7	152.88	0.00262179	1.81788e-06	0.00189403	271.961		
Sample8	950.254	0.00537992	1.20632e-06	0.00134758	377.744		
Sample9	399.878	0.00113963	3.02619e-06	0.0340991	215.557		
Sample10	473.154	0.00138481	6.42891e-06	0.0229761	493.793		
Sample11	286.226	0.00900311	2.69942e-06	0.00238747	181.306		
Sample12	533.384	0.00887295	1.15935e-06	0.00177146	185.647		

Abb. 4.23 Tabelle zur Darstellung der 30 Zufallswerte des Level-E-Modells

Nach der Generierung der Zufallswerte geht man im nächsten Schritt zur Einstellung der Ausgabeparameter über. Die Ausgabeparameter sind schon durch die Modellauswahl vorbestimmt und werden in der Ausgabetable aufgeführt. Die Ausgabeparameter werden in Ausgabedateien hinterlegt und von RESUS eingelesen werden (Siehe Abb. 4.24). Die in der Tabelle einstellbare „Output-Option“ legt fest, ob der jeweilige Parameter in der weiteren Berechnung und Darstellung berücksichtigt werden soll. In unserem Beispiel belassen wir alle Parameter in der Darstellung. Das originale Level-E Programm wurde modifiziert, damit es zusätzliche Zwischenergebnisse an den Interfaces von Quellterm, Geoschicht 1 und Geoschicht 2 ausgibt.

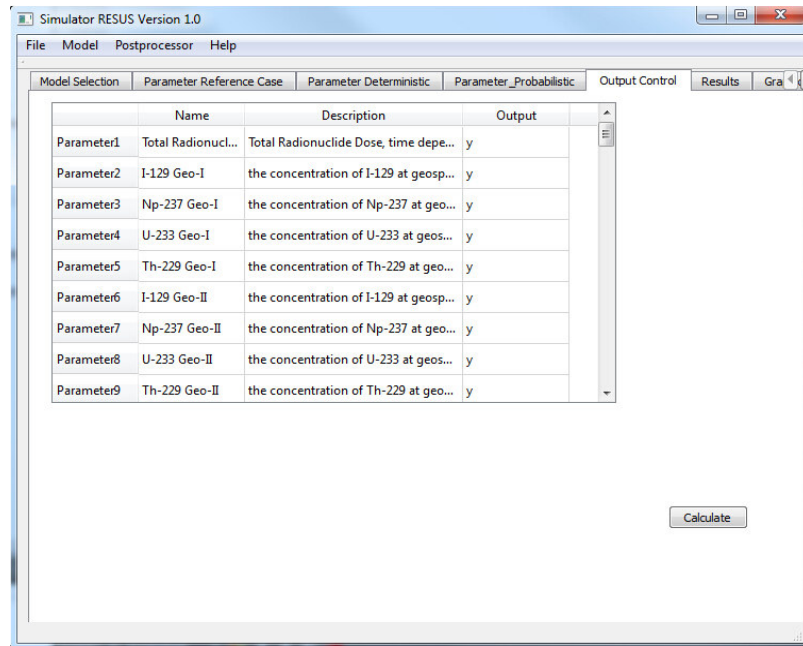


Abb. 4.24 Tabelle für „Output Control“

Nach der Festlegung der zu berechnenden Parameter wird durch einen Mausklick auf den „Calculate“-Knopf das externe kompilierte Berechnungsprogramm „gtmle.exe“ aufgerufen. Es handelt sich bei „gtmle.exe“ um einen kompilierten Level-E Solver, siehe Abb. 4.25. Dieser führt mit dem Referenzwert und den gegebenen 300 probabilistischen Parameterwerten je eine Rechenrealisierung durch. Die Ergebnisse werden dann automatisch an RESUS zur Analyse im Post-Processing weitergegeben.

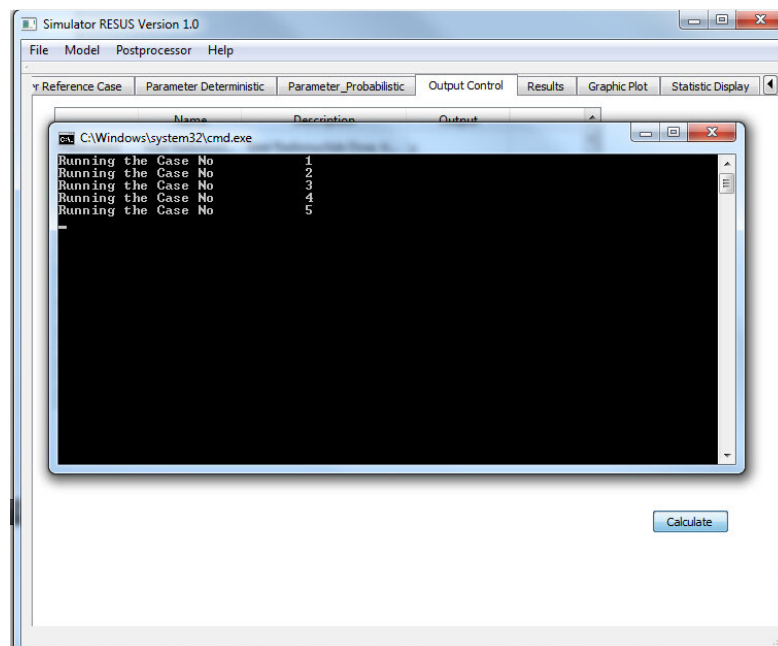


Abb. 4.25 Bildschirm-Ausschnitt beim Aufruf vom Level-E Solver: „gtmle.exe“

Ist die Modellberechnung abgeschlossen, beginnt die Nachbearbeitung und Aufbereitung der Ergebnisse im Post-Processing. Alle Level-E-Ergebnisse sind zeitabhängig, deshalb findet ihre grafische Auswertung im für Zeitverläufe vorgesehenen Tab „Graphic Plot“ statt (Siehe Abb. 4.26). Darin sieht man zunächst ein Auswahlnenü mit allen berechneten Ausgabeparametern. Für die Zeitreihendarstellung der Rechenergebnisse eines Parameters wählt man den gewünschten Parameter aus und betätigt den „Simple Plot“-Knopf (siehe Abb. 4.27). Wenn der Knopf „Delete“ gedrückt wird, werden die auf dem Diagramm dargestellten Zeitverläufe gelöscht, damit die neuen Darstellungen Platz haben können.

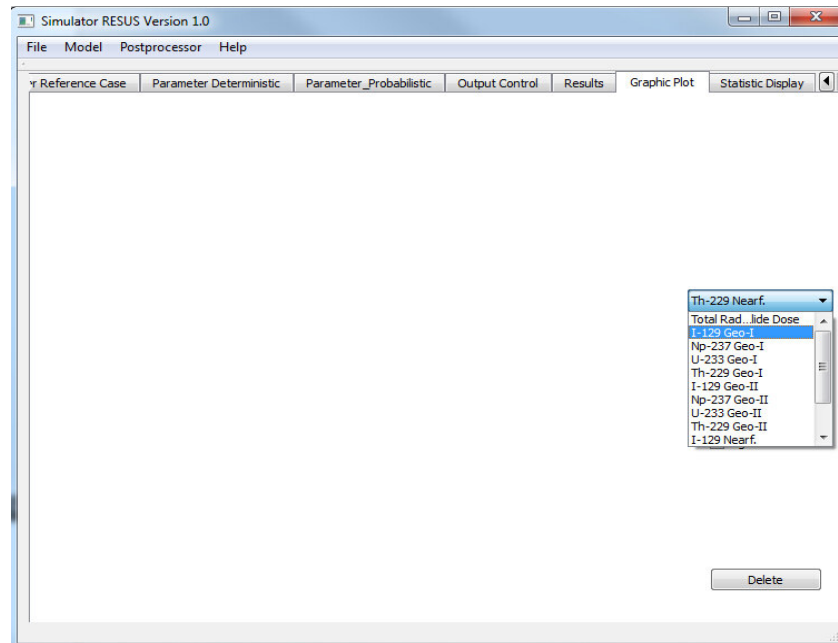


Abb. 4.26 Tab „Graphic Plot“ auf der RESUS-GUI zur grafischen Darstellung der zeitabhängigen Ergebnisdaten



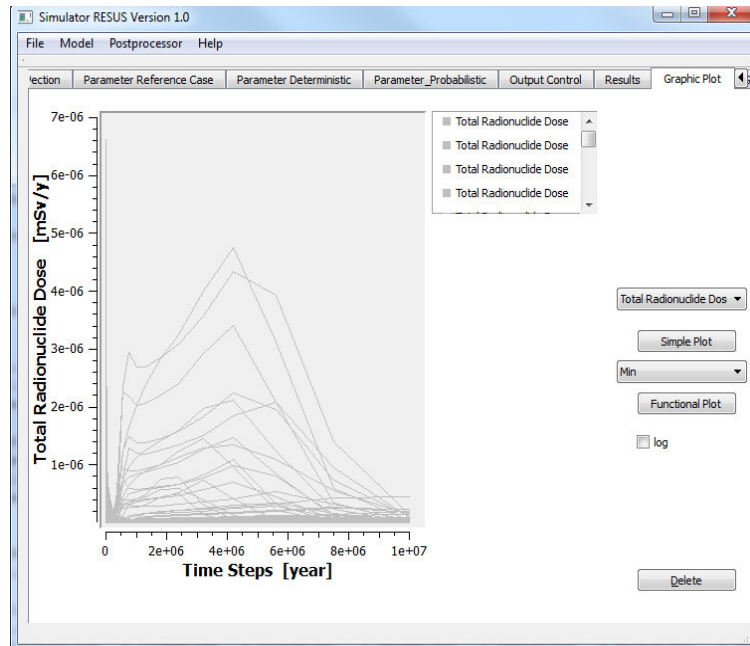


Abb. 4.27 Darstellung von 301 Gesamtdosis-Zeitverläufen, berechnet aus 301 verschiedenen Realisierungen

Auf der Ordinate ist die Dosis in mSv pro Jahr aufgetragen, auf der Abszisse die Zeit in Jahren. RESUS beschriftet die Achsen automatisch mit Name und Einheit des dargestellten Ergebnisparameters. In Abb. 4.27 sieht man, dass die 301 Gesamtdosis-Zeitverläufe aus 301 Rechnungen als graue Linien dargestellt werden.

Die meisten Linienvverläufe bleiben im niedrigen mSv-Bereich. Viele Zeitreihen haben zu Simulationsbeginn einen hohen Sprung in ihrem Zeitverlauf. Danach gibt es häufig noch ein zweites lokales Maximum im Zeitraum von etwa  $2 \cdot 10^6$  bis  $8 \cdot 10^6$  Jahren.

Wenn man die zwei lokalen Maxima mit den Konzentrationsverläufen von I-129 in Abb. 4.28a und die Konzentrationsverläufe von Np-237 in Abb. 4.28b vergleicht, ist zu erkennen, dass der erste Sprung von I-129 verursacht ist, und der zweite Sprung von der Np-U-Th Zerfallskette verursacht ist. Die ist aufgrund der geringen Jod-Sorption plausibel.

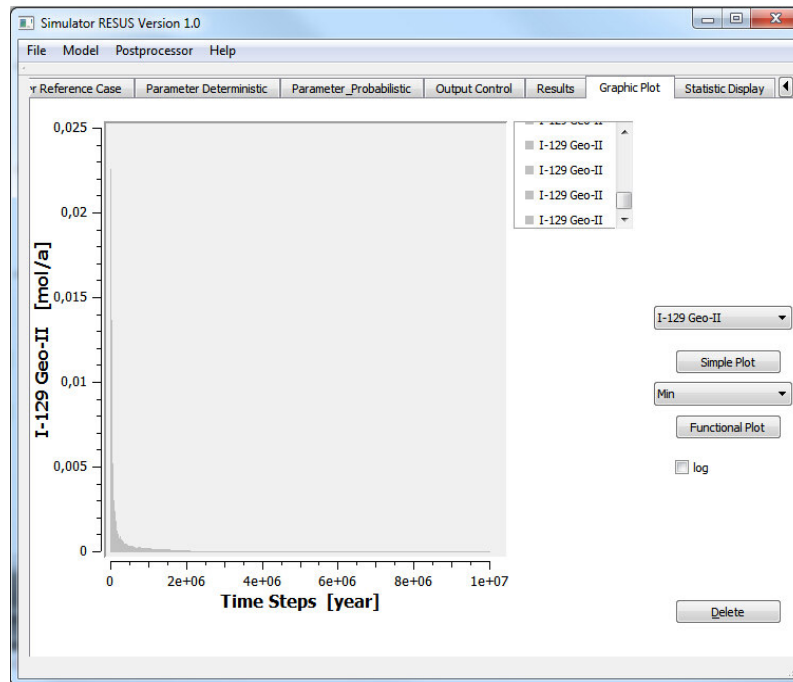


Abb. 4.28a Zeitlicher Verlauf der Konzentration von I-129 am Endpunkt der zweiten Geoformation

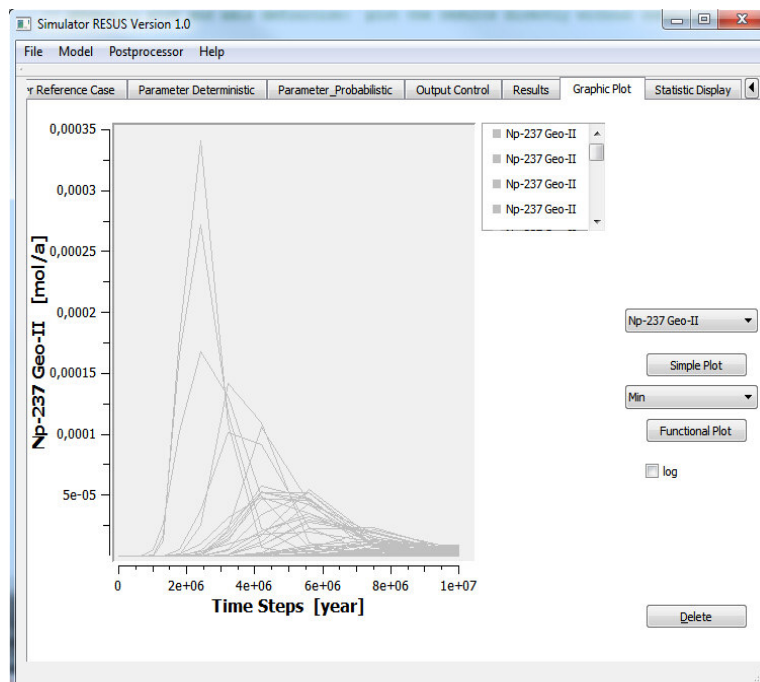


Abb. 4.28b Zeitlicher Verlauf der Konzentration von Np-237 am Endpunkt der zweiten Geoformation

Die Ausbreitung von Np-237 findet im Zeitraum von  $2 \cdot 10^6$  bis  $8 \cdot 10^6$  Jahren statt. Bei manchen Parametern ist die Darstellung mit einer logarithmischen Transformation sinnvoller, z.B. beim

Dosiswert. Abb. 4.29 zeigt die Darstellung der logarithmisch transformierten Dosiswerte im Lauf der Zeit. Wenn die Option „log“ gewählt und die Darstellung mit „Simple Plot“ ausgeführt wird, werden die Ergebnisdaten logarithmisch transformiert dargestellt, wobei die Zeitachse nicht transformiert wird. Da 0-Werte nicht logarithmisch transformiert werden können, werden diese durch den kleinsten transformierten Wert der Datenreihe ersetzt.

Für die statistische Analyse können die statistischen Kennwerte der errechneten Daten über die Zeit dargestellt werden. Diese Quantile sind für eine Sicherheitsanalyse bei Endlagersystemen wichtige Kenngrößen. Für die Darstellung wird das Menü unter dem Knopf „Simple Plot“ gebraucht. Das Menü enthält die Optionen: „Min“ für Minimale Werte; „25% Quantile“ für das 25% Quantil; „Median“ für das 50% Quantil; „75% Quantil“ für das 75% Quantil; „Max“ für das 100% Quantil. „Mean“ für die Mittelwerte. All diese statistischen Kennwerte werden zeitabhängig als Linien im Diagramm dargestellt. In Abb. 4.29 werden alle sechs aufgeführten statistischen Maße für die Dosiswerte dargestellt.

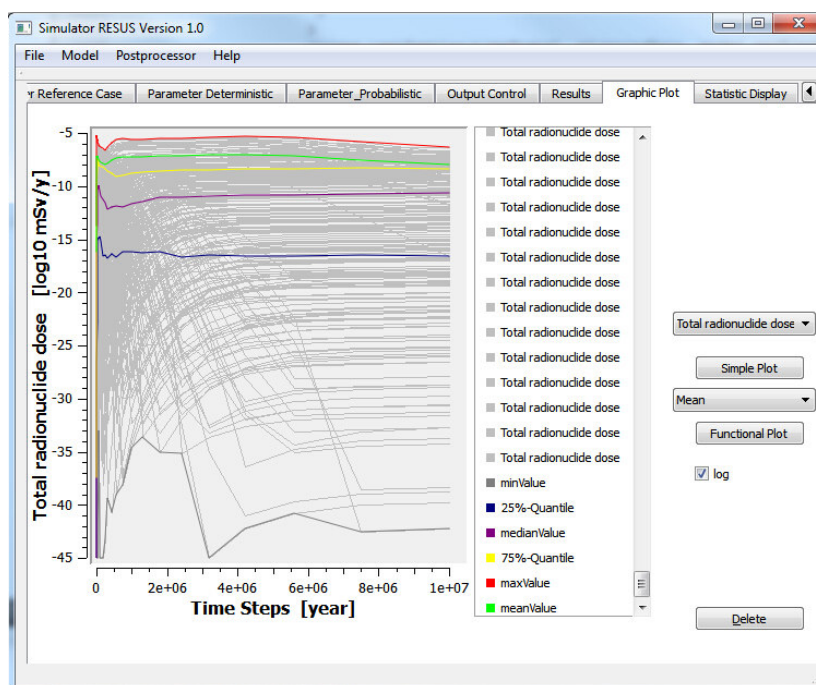


Abb. 4.29 Darstellung der statistischen Maße als dicke Linien über den Realisierungen

Die statistischen Kennwerte der Gesamtdosiswerte werden zusammen mit den Zeitreihenverläufen der Realisierungen dargestellt. Die breite graue Linie gibt die minimale Schadstoffdosis über die Zeit an, die blaue Linie ist das 25% Quantil, die violette Linie ist der Medianwert, die gelbe Linie ist das 75% Quantil, die grüne Linie stellt den zeitlichen Verlauf der Mittelwerte dar, die rote Linie beschreibt die maximalen Werte aller Realisierungen. Die dargestellten Mittelwerte sind die Mittelwerte der originalen Daten, die auch logarithmisch transformiert werden.

Neben der zeitlichen Darstellung sind weitere Analysemethoden in Tab „Statistic Display“ zu finden. Genau wie bei Kapitel 4.1 können Histogramme und Streudiagramme (Scatterplot) auch

beim Level-E-Modell berechnet werden. Im Textfeld rechts oben können wie vorher die statistischen Quantile des gewählten „Quantity of interest“ abgelesen werden (siehe Abb. 4.30).

Beispielhaft wollen wir mit dem Histogramm zuerst die Gesamtdosis analysieren. Als „Quantity of interest“ wählen wir die Höchstwerte der Realisierungen (Max per realization). RESUS bietet an dieser Stelle auch die Möglichkeit an, die Integralwerte über die Simulationszeit darzustellen, damit die aufsummierten Dosiswerte über die Zeit analysiert werden können.

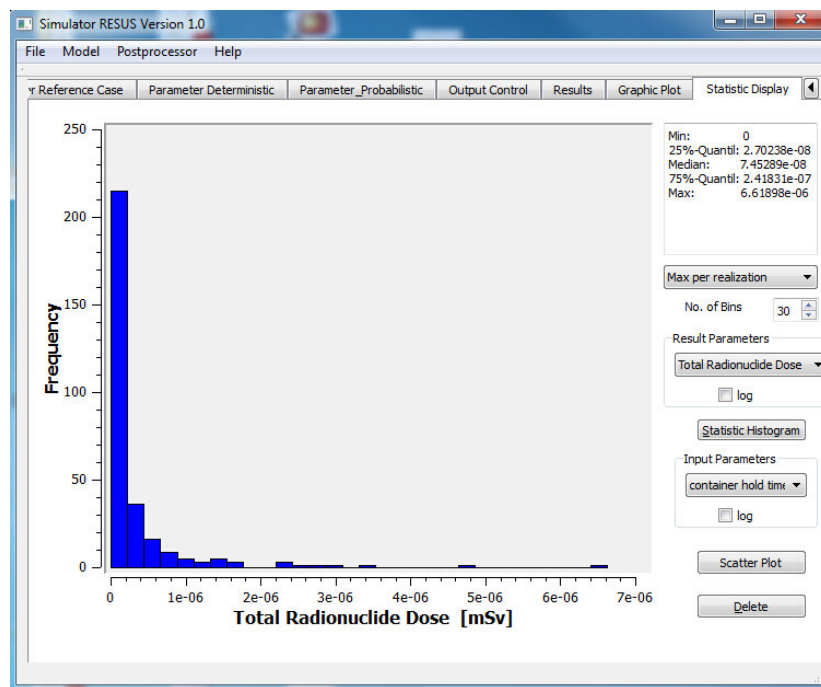


Abb. 4.30 Histogramm der Schadstoffgesamtdosis, 30 Wertklassen

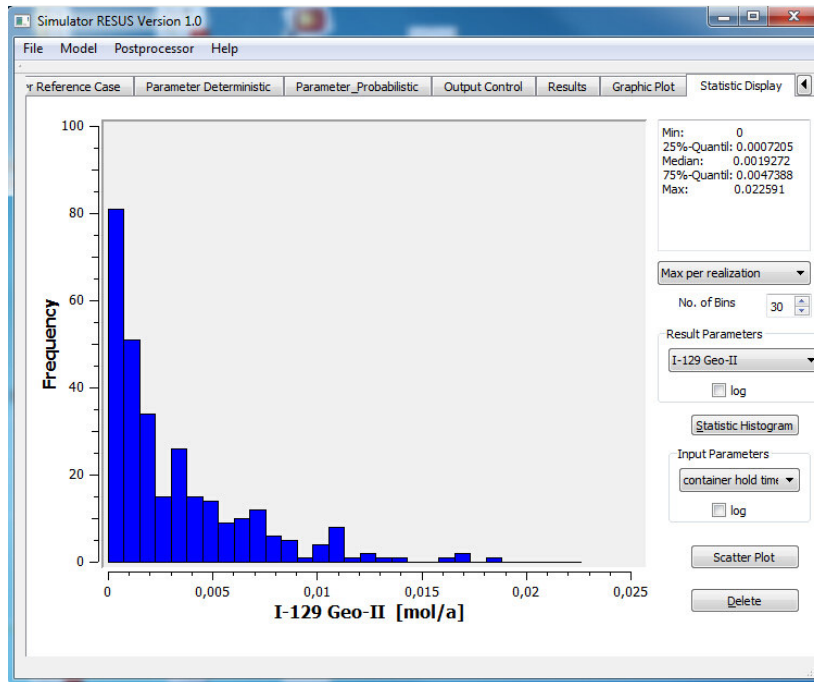


Abb. 4.31 Histogramm des I-129 Konzentrationsflusses am Ende der zweiten Geoschicht, 30 Wertklassen

Das I-129 Nuklid in Abb. 4.31 hat im Vergleich zur den anderen betrachteten Nukliden eine breitere Verteilung. Dies liegt an der geringeren Absorption von I-129 durch die Geoschichten und seiner relativ kurzen Halbwertszeit.

In Abb. 4.32 sieht man, dass die meisten Werte aus dem Konzentrationsfluss in sehr niedrigen Bereichen bleiben, da die Np-Kette relativ stark absorbiert und dadurch der Transport verzögert wird.

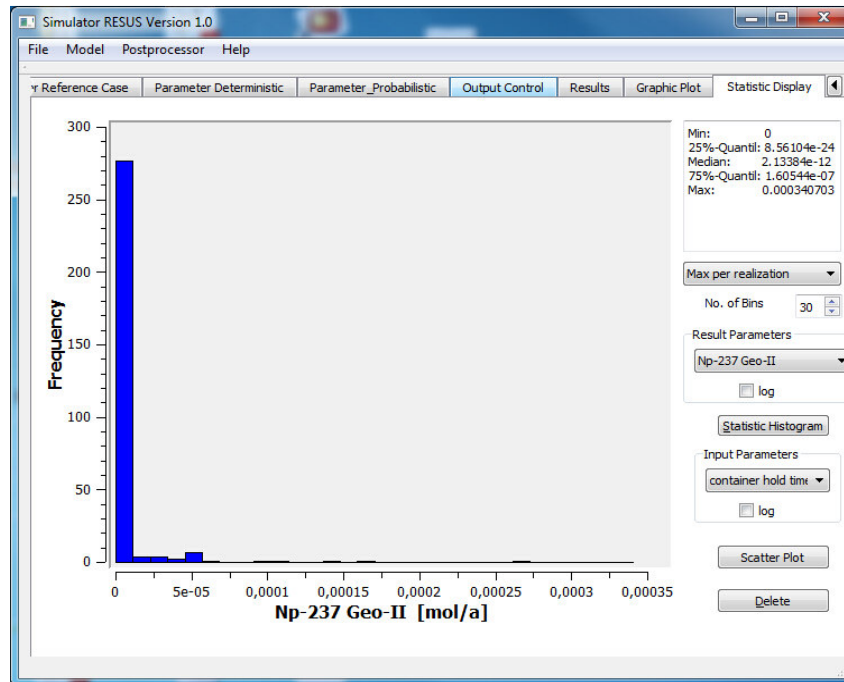


Abb. 4.32 Histogramm des Np-237 Konzentrationsflusses am Endpunkt der zweiten Geoformation, 30 Wertklassen

RESUS bietet die Möglichkeit, die logarithmisch transformierten Daten mit einem Histogramm zu analysieren. Aber wenn die Datenreihe null oder negative Werte enthält, macht die logarithmische Histogrammanalyse keinen Sinn. Daher wird ein Warnungsfenster aufgerufen (siehe Abb. 4.33), um dem Nutzer zu informieren, dass keine logarithmische Transformation mit null oder negativen Werten im Ergebnis für das Histogramm gemacht werden soll.

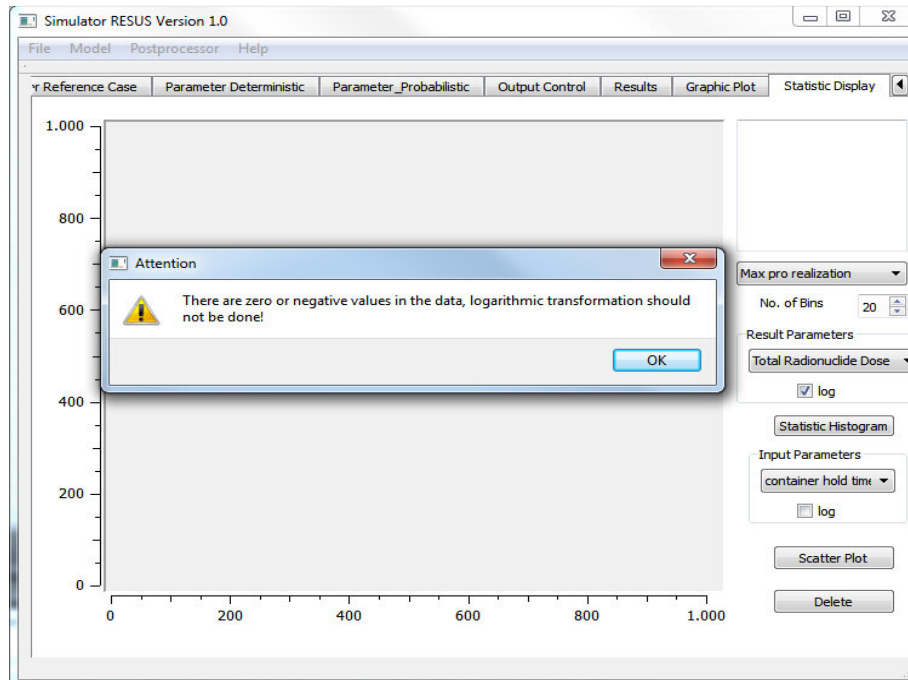


Abb. 4.33 Anzeige einer Warnung, wenn bei der Histogrammanalyse eine logarithmische Transformation mit Nullen oder negativen Werten durchgeführt wird

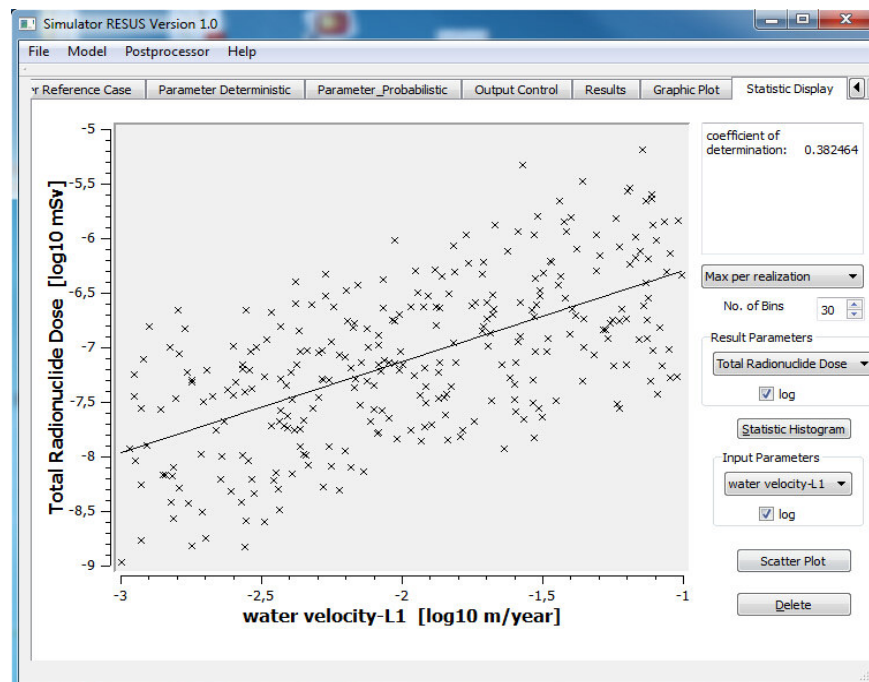


Abb. 4.34 Streudiagramm: logarithmierte Wasserfließrate in der ersten Geoschicht gegen die Gesamtdosis in der Biosphäre

Neben der Auswertung der Histogramme kann vom Benutzer eine Sensitivitätsanalyse mit Streudiagramm (Scatterplot) durchgeführt werden. Unter RESUS ist es möglich, für die

Sensitivitätsanalyse beliebige Ein- und Ausgabeparameter zu wählen. Im folgenden Beispiel werden nur einige der möglichen Kombinationen gezeigt und erläutert. Da null- und negative Werte nicht logarithmisch transformiert werden können, werden diese beim Streudiagramm durch den kleinsten transformierten Wert der Datenreihe ersetzt.

Wie im Textfenster in Abb. 4.34 rechts oben angegeben besteht eine Korrelation mit einem Bestimmtheitsmaß von 0,38. Die Wasserfließrate in der ersten Geoschicht ist folglich einen relativ sensibler Parameter für die Gesamtdosis.

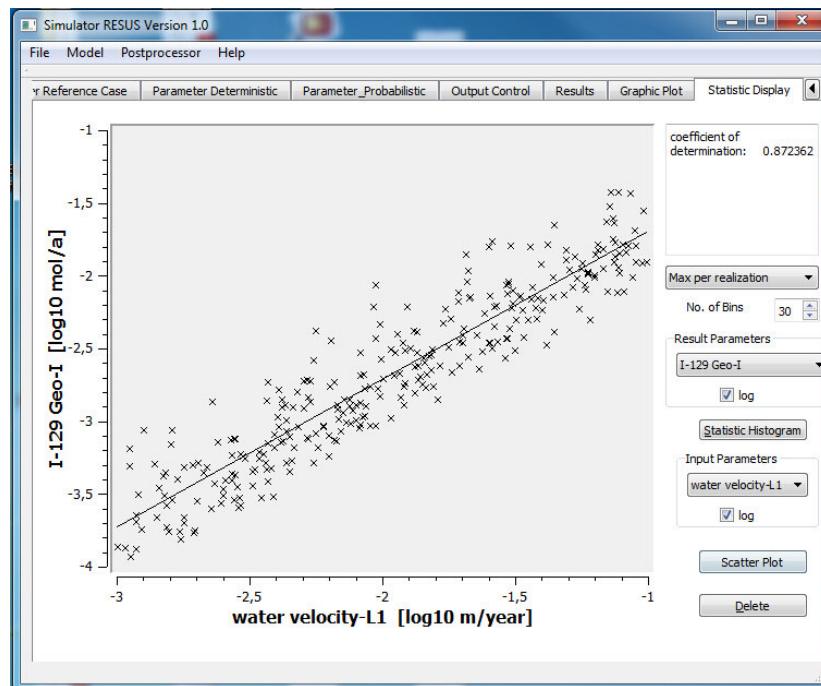


Abb. 4.35 Streudiagramm: logarithmierte Wasserfließrate in der ersten Formation gegen den logarithmierten Konzentrationsfluss von I-129 am Endpunkt der ersten Geoschicht

Es ist naheliegend, dass die Fließgeschwindigkeit in der ersten Geoformation eine entscheidende Rolle für den Ausfluss der Nuklide aus der ersten Formation spielt. Das Bestimmtheitsmaß zwischen diesen Parametern beträgt 0,87, was fast einer linearen Abhängigkeit entspricht.

Der Unterschied zwischen Abb. 4.34 und Abb. 4.35 entsteht möglicherweise daraus, dass die Nuklidkette Np-237, U-233 und Th-229 die Dosiswerte mitbeeinflusst. Die Wasserfließrate spielt beim Transport der Np-Kette wegen deren stärkerer Adsorption eine geringere Rolle. Dieser Effekt wird vom Streudiagramm zwischen der Wasserfließrate und dem Konzentrationsfluss von U-233 in der ersten Geoschicht bestätigt (siehe Abb. 4.36).



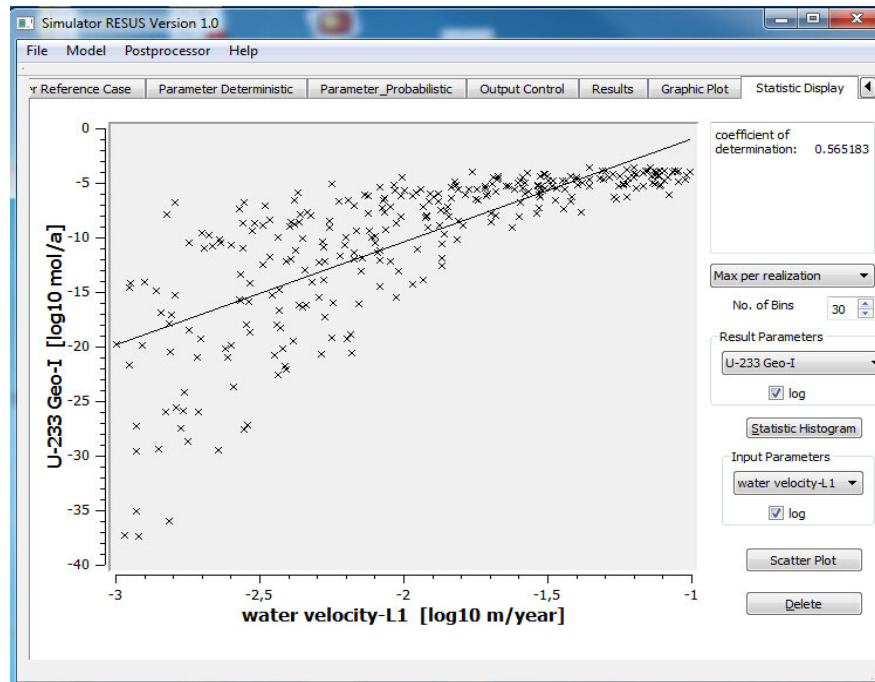


Abb. 4.36 Streudiagramm: logarithmierte Wasserfließrate in der ersten Geoschicht gegen den logarithmierten Konzentrationsfluss von U-233 am Endpunkt der ersten Geoschicht

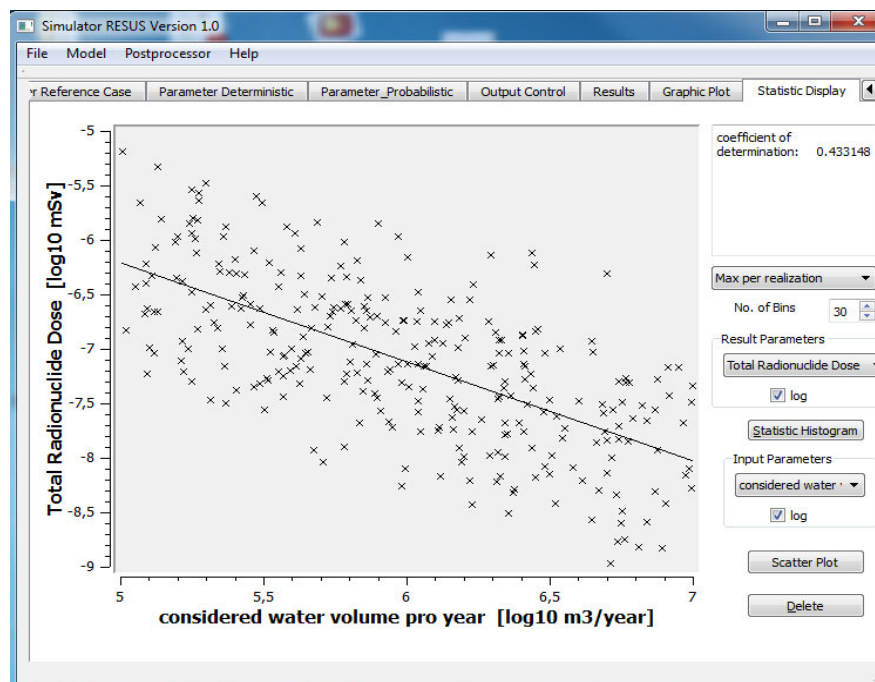


Abb. 4.37 Streudiagramm: logarithmierte Wassermenge in der Biosphäre gegen die logarithmierte Gesamtdosis

Weiter interessiert uns die Korrelation zwischen der Gesamtdosis und dem betrachteten Wasservolumen in der Biosphäre. Beide Parameter sind erneut in einem Streudiagramm aufgetragen (siehe Abb. 4.37). Dieses ist doppelt logarithmisch transformiert und liefert ein Bestimmtheitsmaß von 0,43. Es zeigt, dass das betrachtete Wasservolumen der Biosphäre aufgrund seines Verdünnungseffektes eine negative Wirkung auf die Gesamtdosiswerte hat.

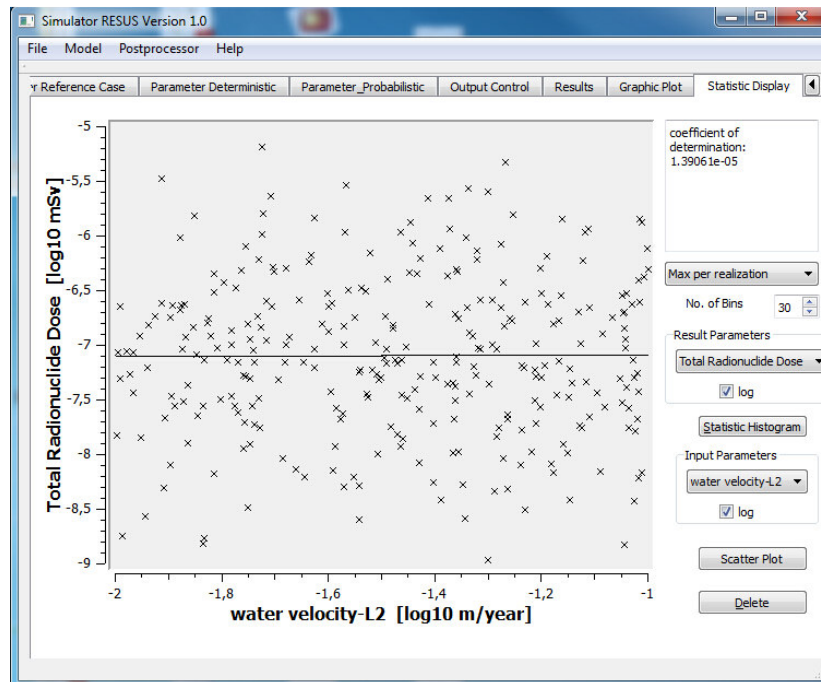


Abb. 4.38 Streudiagramm: logarithmierte Wasserfließrate in der zweiten Geoschicht gegen die logarithmierte Gesamtdosis

Die Auswertung der Abb. 4.38 bringt ein sehr interessantes Ergebnis: Die Korrelation zwischen der Wasserfließrate in der zweiten Geoformation und der Gesamtdosis in der Biosphäre ist schwach. Das Bestimmtheitsmaß beträgt  $1,39 \cdot 10^{-5}$ . Statistisch gesehen besteht keine lineare Beziehung zwischen diesen beiden Parametern. Dies liegt möglicherweise daran, dass die Bedeutung der Wasserfließrate in der zweiten Geoschicht wegen der großen Dominanz der Fließrate in der ersten Geoformation vernachlässigbar ist.

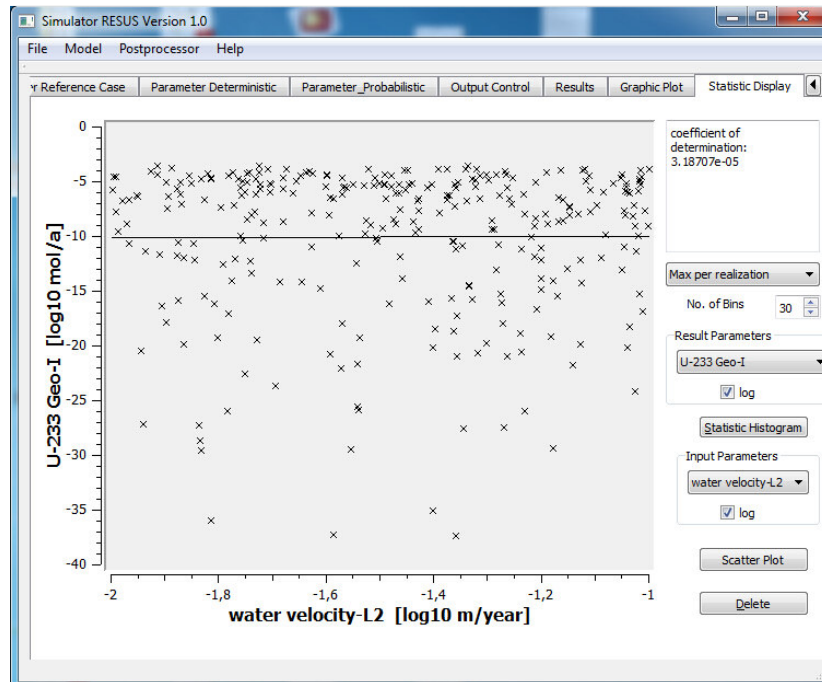


Abb. 4.39 Streudiagramm: logarithmierte Wasserfließrate in der zweiten Geoformation gegen die logarithmierte Ausflussrate von U-233 in der ersten Geoformation

Das Streudiagramm kann auch für eine logische Überprüfung der Modellrechnung genutzt werden. So sollte beispielsweise die Ausflussrate der Nuklide in der ersten Geoformation nicht von der Wasserfließrate in der zweiten Geoformation beeinflusst werden. Ist dies aber doch der Fall, so wurde die Modellrechnung fehlerhaft ausgeführt. Um einen solchen Test auch in unserem Beispiel durchzuführen, tragen wir im nächsten Streudiagramm die Ausflussrate von U-233 gegen die Wasserfließrate in der zweiten Geoformation auf (Siehe Abb. 4.39). RESUS erkennt hier, wie erwartet, keine Korrelation.

## 4.3 Eindimensionales Transportmodell in RESUS

### 4.3.1 Allgemeine Vorstellung eines eindimensionalen Transportmodells in RESUS

In diesem Kapitel soll ein weiteres Modell betrachtet werden. Es heißt „Transportmodell“ und ist geeignet, bestimmte Transportprozesse zu modellieren. In diesen Modellen simuliert man den durch Wasserbewegungen in geologischen Schichten verursachten Transport freigesetzter Radionuklide bis zur Erdoberfläche. Das Transportmodell in RESUS stellt diesen Transport von

Schadstoffen (Radionukliden) in Deck- und Nebengebirge (im Fernfeld) des betrachteten Endlagersystems vereinfacht dar.

Als Deckgebirge bezeichnen Geologen geschichtete Gesteinsbereiche, die auf einem älteren Grundgebirge liegen. Die Endlagerforschung versteht unter dem Begriff meist Gesteinsschichten, die das Wirtsgestein eines Endlagers überlagern. Zum Nebengebirge werden diejenigen Gesteinsschichten gerechnet, die das Wirtsgestein umgeben. Sollten die eingelagerten Radionuklide aus dem Wirtsgestein austreten, werden sie weiter mit dem Grundwasser durch das Deckgebirge zur Biosphäre transportiert [41].

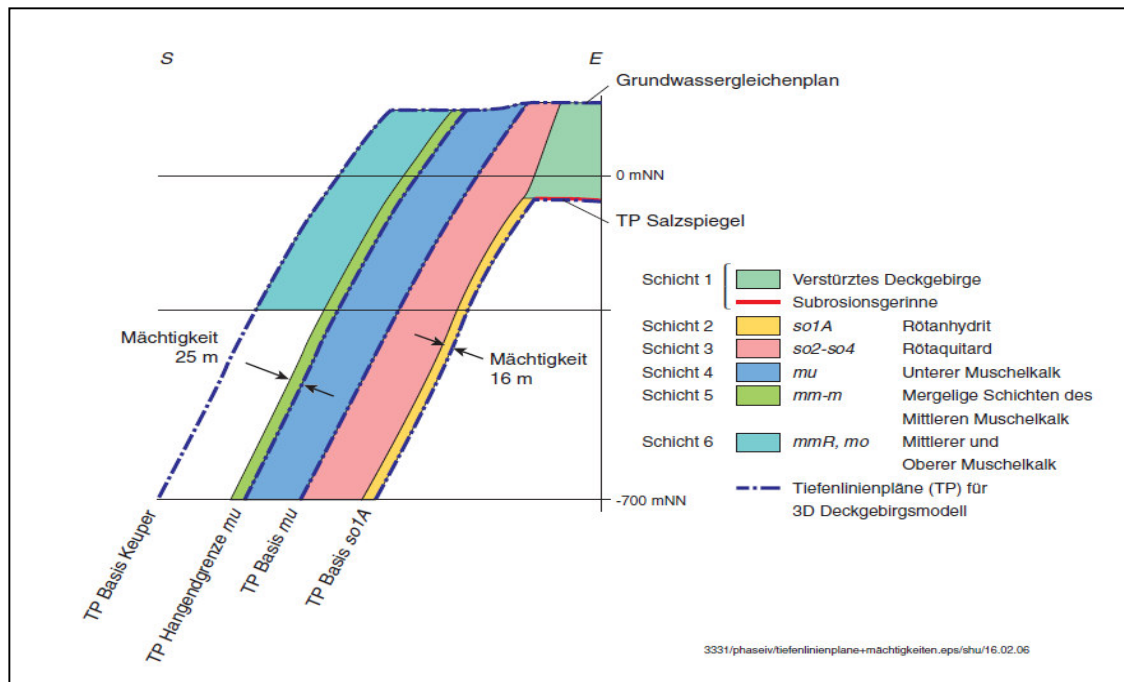


Abb. 4.40 Schematische grafische Darstellung eines Deckgebirges auf einem Salzstock in Deutschland [40]

Das Transportsystem kann als eindimensional, zweidimensional sogar dreidimensional zur Modellierung betrachtet werden. Die eindimensionalen und zweidimensionalen Transportmodelle wurden auf RESUS fest implementiert und können wie auch die vorausgegangenen Beispielmmodelle mit der GUI-basierten Methode verwendet werden.

Die eigentliche Berechnung der Transportprozesse im Deckgebirge wird in diesem Modell, ähnlich wie vorher bei Level-E, mit einem externen Code durchgeführt. Es stehen auf RESUS zwei alternative Rechencodes mit einer Finite-Elemente-Methode zur Auswahl: Zum einen das Rechenmodul aus dem numerischen Code „RockFlow“ [32] und zum anderen dessen Nachfolgeversion „OpenGeoSystem“ (OGS) [26].

Das „RockFlow“-Programm wurde zur numerischen Simulation von Prozessen im Untergrund am Institut für Strömungsmechanik und elektronisches Rechnen im Bauwesen (Heute: Institut für Strömungsmechanik und Umweltphysik im Bauwesen) der Gottfried Wilhelm Leibniz Universität Hannover entwickelt [32]. Für die Simulationsrechnung in RESUS wird die Version

5.1.10 von RockFlow verwendet. OpenGeoSys (OGS) ist eine Weiterentwicklung von RockFlow aus dem Helmholtz Zentrum für Umweltforschung (UFZ). In RESUS wird die Version 5.0.06 genutzt.

Beide Codes arbeiten für den Transportprozess mit den gleichen mathematischen Modellen, daher findet die Transportsimulation in sehr ähnlicher Weise statt. RockFlow hat jedoch die Einschränkung, dass in einem Transportprozess lediglich ein Adsorptionskoeffizient ( $K_d$ -Wert) berücksichtigen werden kann. Modelle mit mehreren geologischen Transportmedien können hier nicht berechnet werden. Bei OGS hingegen sind Modellsimulationen mit verschiedenen Medien und unterschiedlichen Adsorptionskoeffizienten möglich.

RESUS bietet dem Benutzer vorgefertigte eindimensionale und zweidimensionale Transportmodelle mit unterschiedlichen Modellierungsmöglichkeiten an. Wählbar sind beispielsweise beliebig viele geologische Schichten, Radionuklide und Adsorptionsisothermen.

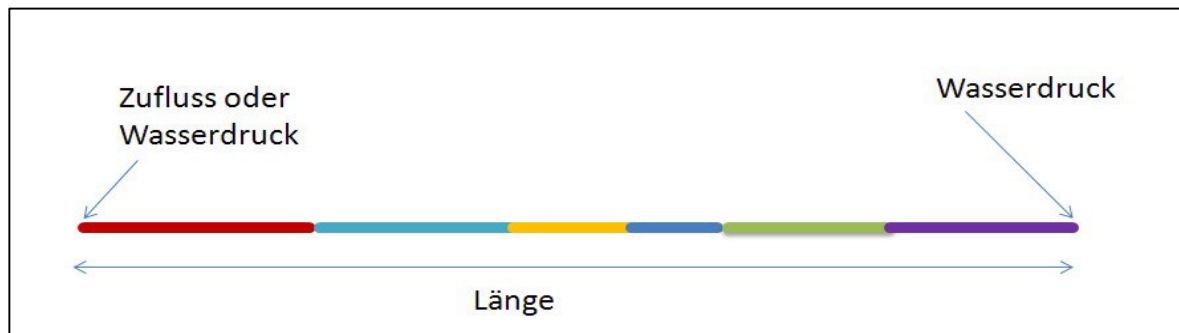


Abb. 4.41 Eindimensionales Transportmodell eines Deckgebirges als ein Beispiel

Das Deckgebirge in diesem Beispiel besteht aus sechs unterschiedlichen geologischen Schichten. Durch diese werden die Schadstoffe mit dem Grundwasser transportiert. Die Transportstrecke wird hier zu einer eindimensionalen Linie vereinfacht. Die berechnete Größe ist die Schadstoffkonzentration am Ende der Strecke. Die hydraulischen Randbedingungen und Materialeigenschaften der geologischen Schichten werden von RESUS als deterministische oder probabilistische Parameter festgelegt. Diese werden in den ein- und zweidimensionalen Transportmodelle in den folgenden Gleichungen verwendet [30].

$$\theta \frac{\partial C}{\partial t} + \rho \frac{\partial S}{\partial t} + q_a \frac{\partial C}{\partial t} = \theta \frac{\partial}{\partial x_a} \left( D_{xy} \frac{\partial C}{\partial x_b} \right) - \lambda \theta C - \lambda S \rho$$

$$D_{\alpha\beta} = (\alpha_T |v| + D^*) \delta_{\alpha\beta} + (\alpha_T + \alpha_L) \frac{v_a v_\beta}{|v|} \quad (4-8)$$

Dabei ist  $t$  die Zeit [s],  $x$  ist der Transportweg [m],  $C$  steht für die Schadstoffkonzentration in der gelösten Phase [kg/kg],  $S$  ist die Stoffkonzentration der vom Gestein absorbierten Phase [kg/kg],  $\rho$  ist die Dichte des Gesteins [kg/m<sup>3</sup>],  $\theta$  ist die Porosität des Gesteins [-],  $D_{\alpha\beta}$  bezeichnet den hydrodynamischen Dispersionstensor [-],  $q_a$  ist die Darcy-Geschwindigkeit der Grundwasserströmung [m/a],  $\lambda$  ist die Zerfallskonstante des betrachteten Radionuklids [1/s],  $D^*$

ist der effektive Diffusionskoeffizient [m<sup>2</sup>/s],  $\alpha_L$  bezeichnet die longitudinale Dispersionslänge [m],  $\alpha_T$  ist die transversale Dispersionslänge [m].

Bei diesem mathematischen Modell werden der advective Stofftransport, die Dispersion, die molekulare Diffusion und der Adsorptionseffekt in den porösen Medien berücksichtigt. Die Wasserströmung wird nach dem Darcy-Gesetz, Massenerhaltungsgleichung und der resultierenden Richards-Gleichung berechnet:

$$q_a = -k/u \nabla P \quad (4-9)$$

$$\frac{\partial}{\partial t} \theta \cdot S \cdot -div \frac{k}{u} \nabla P = 0 \quad (4-10)$$

$k$  [m<sup>2</sup>] ist die Permeabilität der Geoschicht und  $\nabla P$  [-] ist der Gradient vom Wasserdruck,  $S$  [-] bezeichnet den Wassersättigung am Porenraum.

Für die Gleichgewichtssorption wird eine algebraische Beziehung zwischen der Stoffkonzentration der vom Gestein absorbierten Phase  $S$  [kg/kg] und der Schadstoffkonzentration in der gelösten Phase  $C$  [kg/kg] aufgestellt:

$$S = f(C) \quad (4-11)$$

Tabelle 4-3 Lineare/Nichtlineare Adsorptionsisotherme [30]

Phänomenologischer Ansatz	Isotherme, Model	Referenz
$S = k_1 \times C$	Henry	Lapidus und Amundson (1952)
$S = k_1 C k_2$	Freundlich	Freundlich (1909)
$S = \frac{k_1 C}{1 + k_2 C}$	Langmuir	Langmuir (1918)
$S = \frac{k_1 C^{k_3}}{1 + k_2 \times C^{k_3}}$	Freundlich-Langmuir	Sips(1950)
$S = \frac{k_1 C^{k_3}}{1 + k_2 \times C^{k_3} + k_3 \sqrt{C}}$	Gunary	Gunary(1970)
$S = k_1 (1 - [1 + k_2 C^{k_3}]^{k_4})$	Barry	Barry(1992)

Die Sorptionsisotherme, die RockFlow und OGS berücksichtigen, sind in Tabelle 4-3 zusammengestellt: Dabei sind  $k_1$ ,  $k_2$ ,  $k_3$  und  $k_4$  stoff- und bodenspezifische Parameter, die z.B. abhängig vom pH-Wert und von den Kationenaustauschkapazitäten sind. RESUS bietet diese sechs Gleichungen in der GUI-basierten Methode zur Auswahl an.

#### 4.3.2

#### Ein Beispiel für eine eindimensionale Transportmodellierung

In diesem Unterkapitel wird ein Beispielmmodell vorgestellt, welches vom Verfasser der vorliegenden Arbeit im Rahmen des Vorhabens „Mitarbeit bei der Prüfung einer längerfristigen Offenhaltung der Asse unter dem Aspekt möglicher Konsequenzen eines auslegungsüberschreitenden Lösungszutritts während des Offenhaltungsbetriebs“ erstellt worden ist. Dieses Modell ist hier von Interesse, da es den für die RESUS-Modellierung relevante Transportprozess abbildet. Es geht darin um die Beurteilung der radiologischen Risiken im Fall eines auslegungsüberschreitenden Lösungszutritts bzw. „Absaufens“ der Schachanlage Asse II.

In der Schachanlage Asse II wurde von 1909 bis 1925 Kalisalz und von 1916 bis 1964 Steinsalz abgebaut. Am 31.03.1964 wurde die Salzförderung eingestellt. Der stillgelegte Schacht wurde 1965 vom Bund gekauft und nach einer ersten Versuchsphase zur Einlagerung radioaktiven Abfalls genutzt. Von 1967 bis 1978 wurden insgesamt 125.787 Fässer und Gebinde mit schwach- und mittelradioaktiven Abfällen in der Asse II eingelagert [36].

Der hohe Durchbauungsgrad und die Nähe der Abbaukammern zum Nebengebirge sind heute Ursache immer größerer Probleme in der Asse. Unter dem Druck des Deckgebirges, der auf dem Grubengebäude lastet, geben die Abbaukammern nach. Dadurch werden sowohl das Salz- als auch das Nebengebirge aufgelockert. Es entstehen Klüfte durch die Grundwasser in die Grube eindringt. Diese sogenannten Zutrittswässer sind derzeit salzgesättigt. Sie dringen schon seit 1988 vom oberen Teil der Südflanke, in etwa 500 bis 575 Meter Tiefe, in das Bergwerk ein. Hinzu kommt, dass die Abbaukammern durch die Gebirgsbewegung instabil werden. Teilweise sind die Deckenzwischen den Kammern bereits eingebrochen. Größere unkontrollierte Grundwassereinbrüche sind zu befürchten [36].

Zu dieser Problematik hat das Bundesamt für Strahlenschutz (BfS) als heutiger Betreiber der Asse drei Lösungsoptionen untersuchen lassen. Diese sind: Vollverfüllung des Grubengebäudes, Umlagerung der Abfälle in eine tiefere Ebene und Rückholung der Abfälle [36].

Vor diesem Hintergrund wurde im Rahmen des o.g. Vorhabens das hier betrachtete Modell für den Transportprozess im Deckgebirge der Asse unter Annahme einer längerfristigen Offenhaltung des Grubengebäudes aufgebaut und analysiert. Es handelt sich dabei um ein eindimensionales Transportmodell. Darin sind die folgenden Aspekte des Schadstofftransports berücksichtigt:

1. Um den Rechenaufwand zu reduzieren, wurde nach Rücksprache mit dem Projektpartner für das Asse-Beispiel der Transportpfad auf eine geologische Schicht im Modell abgebildet.
2. Der Transportweg wird als eindimensionales poröses Medium betrachtet. Auf beiden Seiten können Randbedingungen angegeben werden.

2. Auf beiden Seiten wird der hydraulischen Druck bzw. der Wasserzufluss in den Randbedingungen festgelegt. Dieser soll am Einstromrand (in Abb. 4.42 links) höher sein als am Ausstromrand.

3. Am Einstromrand wird eine Randbedingung für die Schadstoffkonzentration (Nuklid-Konzentrationen) festgelegt. Der Schadstofftransport findet advektiv mit der Wasserströmung statt. Das Modell berücksichtigt zudem die diffusive Ausbreitung.

4. Die Adsorption und die Desorption der Schadstoffe im porösen Medium werden mitgerechnet, wobei die Barriere-Funktion des Deckgebirges berücksichtigt wird.

Der zeitliche Konzentrationsverlauf der Schadstoffe am Ausstromrand ist das Ergebnis dieser Rechnungen.

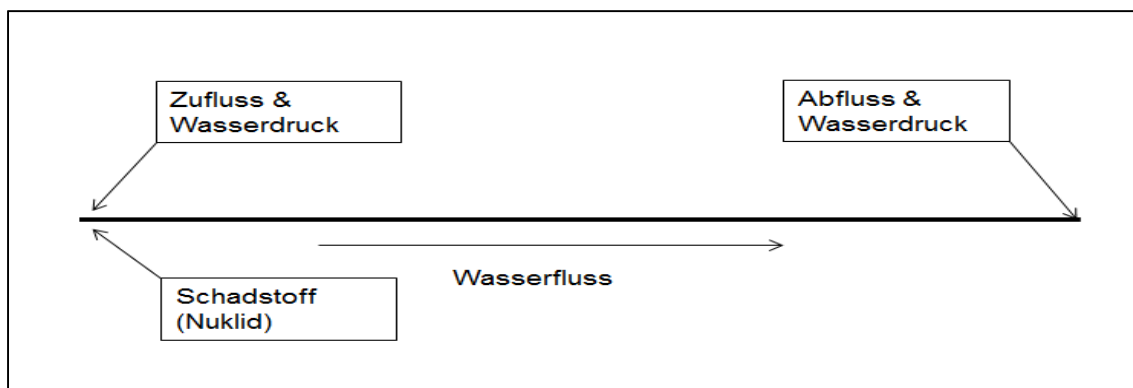


Abb. 4.42 Grafische Darstellung des eindimensionalen Transportmodells in RESUS



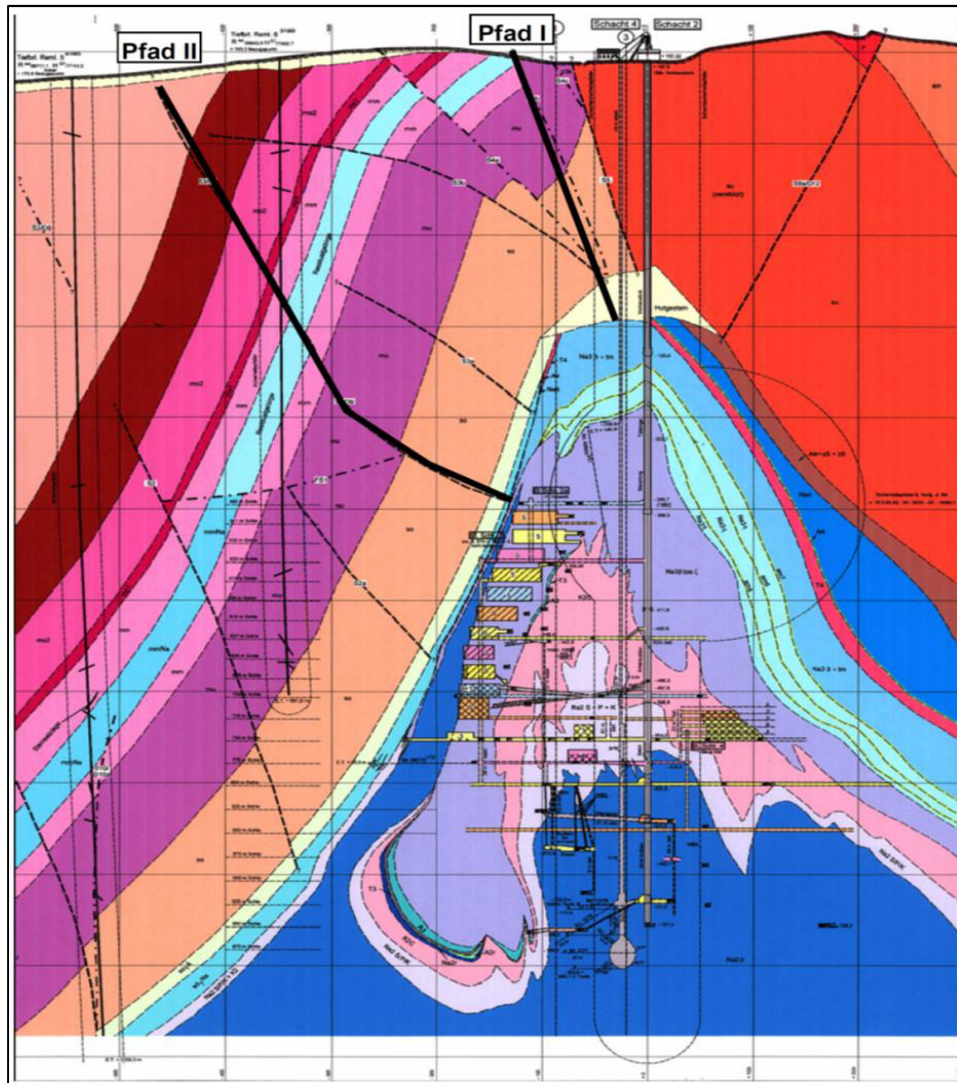


Abb. 4.43 Deckgebirge des Bergwerks Asse mit zwei möglichen Transportpfaden (Pfad I und II) vom Grubengebäude bis zur Erdoberfläche [42]

Weiterhin gelten für das Asse-Modell folgende spezielle Annahmen (siehe Abb. 4.43):

Es gibt in diesem Modell zwei mögliche Transportpfade. Um die Modellierung zu erleichtern werden diese stark vereinfacht. Beide durchlaufen im Modell dieselbe Geoformation (Schicht).

Die kontaminierte Lösung wird aus dem Grubengebäude in einen Transportweg gedrückt. Die Länge dieses Transportweges ist ungewiss und wird daher mit einem probabilistischen Parameter dargestellt.

Der Volumenstrom wird entsprechend den Ergebnissen vorheriger Simulationen zum Konvergenzverhalten und aus Literaturdaten festgelegt. Er ist von der Konvergenzrate des Grubengebäudes abhängig. Der Volumenstrom am Einstromrand wird als probabilistischer Parameter eingetragen.

Der Wasserdruck auf der Seite der Erdoberfläche ist durch den Atmosphärendruck (1 bar) gegeben, da angenommen wird, dass der Pfad die Erdoberfläche erreicht.

Die Adsorption und die Desorption werden mit dem Ansatz nach Henry (lineares Gleichgewicht,  $K_d$ -Wert siehe Tabelle 4-3) berücksichtigt.

Der advective und der diffusive Transport werden gleichzeitig betrachtet.

Die Zeitdauer für die Freisetzung der gesamten kontaminierten Lösung wird durch den Ausfluss-Volumenstrom und das Gesamtvolumen der Lösung im Grubengebäude festgelegt:

Zeitdauer für Freisetzung = Gesamtvolumen Lösung / Ausfluss-Volumenstrom

Es wird angenommen, dass das gesamte Volumen der kontaminierten Lösung einen konstanten Wert von 1.300.000 m<sup>3</sup> hat.

Im folgenden Abschnitt wird die Vorgehensweise zur Erstellung und Berechnung eines eindimensionalen Transportmodells mit RESUS beschrieben. Wie bei den vorherigen Beispielen muss auch beim eindimensionalen Transportmodell zunächst der Modelltyp im ersten „Model Selection“-Tab der RESUS-Oberfläche gewählt werden (siehe Abb. 4.44). Unter der Überschrift „Geosphere Transport“ sucht man die Auswahl „1-D\_RockFlow“. Beim Menü kann man unterscheiden mit welchem Programm das Modell berechnet werden soll. Da für dieses Modell beide Programme (RockFlow und OGS) die gleichen Ergebnisse liefern, zeigen wir in dieser Dissertation das Transportmodell, das mit RockFlow berechnet wird. Mit einem Doppelklick auf „1-D\_RockFlow“ öffnet sich ein zusätzliches Fenster, in dem die Anzahl der im Modell betrachteten Geoformationen und Nuklide festgelegt werden. Wie oben erläutert, wird für das Asse-Beispiel lediglich eine geologische Schicht im Modell berechnet. Die Einstellung der Anzahl der Geoformationen wird im Beispiel folglich auf „1“ gesetzt.

Wenn die eingegebene Anzahl der Nuklide größer als 1 ist, erscheint neben dem Eingabekästchen die Option „Decay Chain“. Mit dieser Option kann der Nutzer entscheiden, ob die in der Simulation betrachteten Nuklide über eine Zerfallskette verbunden werden.

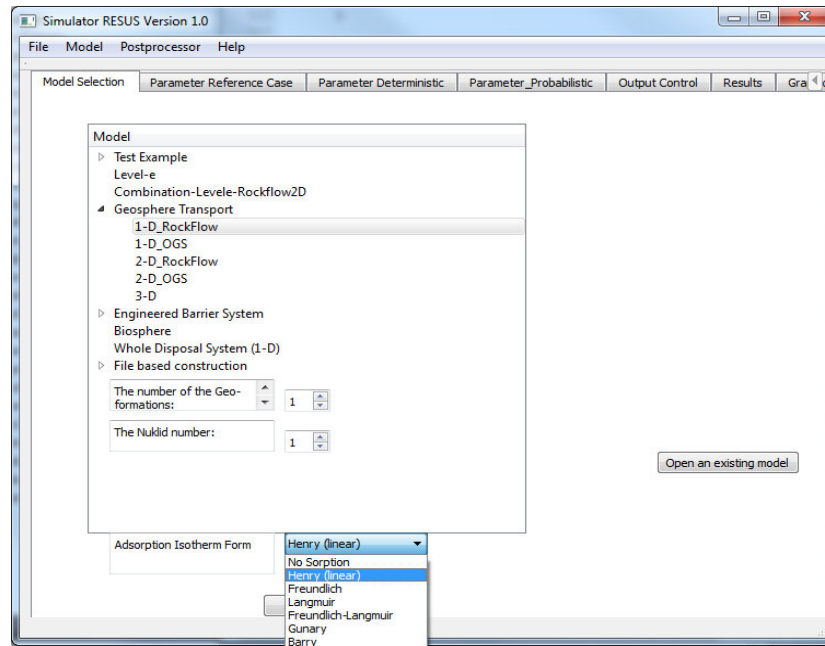


Abb. 4.44 Das erste Tab der Benutzeroberfläche für die Modellauswahl eines eindimensionalen Transportmodells

Weiter unten im Fenster befindet sich das Auswahlmenü zur Modellierung der Sorption. Die RockFlow und OGS-Transportmodelle bieten hier sechs Möglichkeiten an. Diese wurden in Tabelle 4-3 beschrieben.

Für das Asse-Beispiel werden elf kritische Isotope betrachtet: Cl-36, Ni-59, Se-79, Tc-99, Ni-63, Sr-90, Cs-137, Co-60, Pu-239, U-235 und Pa-231, wobei Pu-239, U-235 und Pa-231 zusammen eine Zerfallskette bilden. Die Nuklide unterscheiden sich stark in ihren Halbwertszeiten. Für die Berechnung bietet sich eine entsprechende Aufteilung in vier Zerfalls-Gruppen an:

Gruppe 1: Co-60 - extrem kurzlebige Nuklide

Gruppe 2: Ni-63, Sr-90, Cs-137 - kurzlebige Nuklide

Gruppe 3: Cl-36, Ni-59, Se-79 Tc-99 - langlebige Nuklide

Gruppe 4: Pu-239, U-235 und Pa-231 als eine Zerfallskette

Die Modellrechnung wird für jede Gruppe separat durchgeführt. In dieser Dissertation wird die Gruppe 4 mit Pu-239, U-235 und Pa-231 genauer betrachtet.

Für die Erstellung des Asse-Modells setzten wir im ersten Tab die Anzahl der Geoformationen auf „1“ und die Anzahl der Nuklide auf „3“. „Decay Chain“ wird ausgewählt. Für das Asse-Beispiel wird für alle betrachteten Radionuklide das Henry'sche  $K_d$ -Konzept verwendet, also im Sorptions-Menü „Henry (linear)“ gewählt. Mit einem Klick auf den Knopf „OK“ wird das Modell erzeugt (siehe Abb. 4.44).

Wie bei den vorangegangenen Modellen werden im nächsten Schritt die Referenzwerte für die probabilistischen Parameter festgelegt. Der Parametersatz variiert je nach Einstellung der Nuklidanzahl, Geoformationen und Adsorptionsoptionen. Für dieses Beispiel sind die probabilistischen Parameter in der folgenden Tabelle 4-4 aufgelistet.

Tabelle 4-4 Einstellung der Referenzwerte für die probabilistischen Parameter

Parameter	Einheit	Referenzwert
Porosität	-	0,15
Dispersionslänge	m	10
Speicherkoeffizient	-	0
Permeabilität	m <sup>2</sup>	10 <sup>-12</sup>
Gesteinsdichte	kg/m <sup>3</sup>	2700
Formationslänge	m	300
Diffusionskoeffizient	m <sup>2</sup> /s	10 <sup>-5</sup>
Freisetzungskonzentration von Pu-239	kg/m <sup>3</sup>	1,52·10 <sup>-5</sup>
Sorptionskoeffizient K <sub>d</sub> von Pu-239	kg/kg	2,0
Freisetzungskonzentration von U-235	kg/m <sup>3</sup>	0,000502
Sorptionskoeffizient K <sub>d</sub> von U-235	kg/kg	0,0005
Freisetzungskonzentration von Pa-231	kg/m <sup>3</sup>	1,74·10 <sup>-9</sup>
Sorptionskoeffizient K <sub>d</sub> von Pa-231	kg/kg	4,0
Wasserflussrate	m <sup>3</sup> /a	0,5
Fluiddichte	m <sup>3</sup> /kg	1300
Fluidviskosität	Pa×s	0,001
Druck am Anfang des Transportpfads	Pa	0
Druck am Ende des Transportpfads	Pa	20000
Zeitpunkt des Freisetzungsanfangs	Jahr	0,0
Zeitpunkt des Freisetzungsendes	Jahr	0,0
Gesamtvolumen der Lösung	m <sup>3</sup>	1,3·10 <sup>-9</sup>

Parameter	Einheit	Referenzwert
Zeitschritt der Simulation	s	$1,57 \cdot 10^{-9}$
Anzahl der Zeitschritte der Simulation	-	3000
Länge der Geometrieelemente (FEM)	m	1,0
Zerfallskonstante von Pu-239	$s^{-1}$	$9,12 \cdot 10^{-13}$
Zerfallskonstante von U-235	$s^{-1}$	$3,14 \cdot 10^{-17}$
Zerfallskonstante von Pa-231	$s^{-1}$	$6,67 \cdot 10^{-13}$
Maximale Anzahl der Iterationen für Druckrechnung	-	1000
Toleranz von Iteration für Druckrechnung	-	$10^{-12}$
Maximale Anzahl der Iteration für die Berechnung der Stoffkonzentration	-	1000
Toleranz von Iteration für die Berechnung der Stoffkonzentration	-	$10^{-12}$

Die Parameter in Tabelle 4-4 sind die Parameter, die bei der Simulation als probabilistische Parameter betrachtet werden. Bei diesem Modell sind die folgenden Punkte zu beachten:

Wenn ein Wert für den Volumenstrom festgelegt wird, findet der „Wasserdruck am Anfang“ keine Berücksichtigung. Die Wasserströmung richtet sich in diesem Fall direkt nach dem eingegebenen Volumenstrom, dabei wird das Wasserdruckfeld nach dem Darcy'schen Gesetz zurückgerechnet. Wird kein Wert für den Volumenstrom angegeben, so wird das Wasserdruckfeld nach den Druckrandbedingungen gerechnet, die Wasserströmung wird dann mit dem Druckfeld nach dem Darcy'schen Gesetz berechnet.

Ähnlich wie bei dem Volumenstrom ist es bei der Festlegung des Gesamtvolumens. Wird ein Gesamtvolumenwert vorgegeben, so wird der eventuell ebenfalls vorgegebene „Zeitpunkt des Freisetzungsendes“ nicht mehr berücksichtigt. Anstelle dessen wird die Zeitdauer der Freisetzung nachfolgender Gleichung berechnet:

$$\text{Zeitdauer der Freisetzung} = \text{Gesamtvolumen} / \text{Volumenstrom}$$

Der „Zeitpunkt des Freisetzungsendes“ wird dann aus dem Anfangszeitpunkt und der Zeitdauer der Freisetzung bestimmt.

Der Parameter „Zeitschritt“ bestimmt die in einem Simulationsschritt betrachtete Zeit, auch Schrittweite genannt. Bei diesem Beispiel soll ein Zeitschritt  $1,57 \cdot 10^{-09}$  Sekunden, also 50

Jahren betragen. Weiterhin wird die Anzahl der zu simulierenden Schritte auf 3000 festgelegt. Unser Modell berechnet folglich die Entwicklung des Transportprozesses über einen Zeitraum von  $3000 \times 50 \text{ Jahre} = 150000 \text{ Jahre}$ . Die Iterationsschritte für die Berechnung werden durch RockFlow und OGS automatisch geregelt.

Die Zerfallskonstanten der betrachteten Radionuklide sind ein weiterer festzulegender Parameter.

Neben den Zerfallskonstanten müssen im selben Schritt auch Kontrollparameter für die Finite-Elemente-Methode-Simulation festgelegt werden. Dazu gehören die maximale Anzahl der Iterationen bei jedem Zeitschritt und die Toleranz der Fehlerwerte in jeder Iteration. Diese Kontrollwerte gelten dann für alle Rechenrealisierungen [33].

Nach der Eingabe der Referenzwerte für alle probabilistischen Parameter muss der Benutzer den Knopf „OK“ drücken, um die Eingabe zu bestätigen. Wie bereits bei dem vorherigen Beispielmmodell werden an dieser Stelle wieder die Eingaben auf Konsistenz überprüft.

Nach der Festlegung der Referenzwerte der Parameter geht RESUS zum nächsten Tab „Parameter\_Probabilistic“ über. Dort werden die probabilistischen Parameter durch Angabe ihrer statistischen Verteilungen samt Verteilungsmaßen und die anschließende Generierung von Zufallswerten bestimmt. RESUS bietet zudem die Möglichkeit, ausgewählte vom Modell vorgegebene probabilistische Parameter als konstante deterministische Parameter festzulegen. Dafür muss die Option „Deterministic“ gewählt werden (siehe Abb. 4.45). Von dieser Option machen wir auch für einige Parameter in unserem Asse-Modell Gebrauch. Ist die Verteilung des probabilistischen Parameters mit „Deterministic“ festgelegt, wird der Referenzwert des Parameters für alle Rechenrealisierungen eingesetzt.

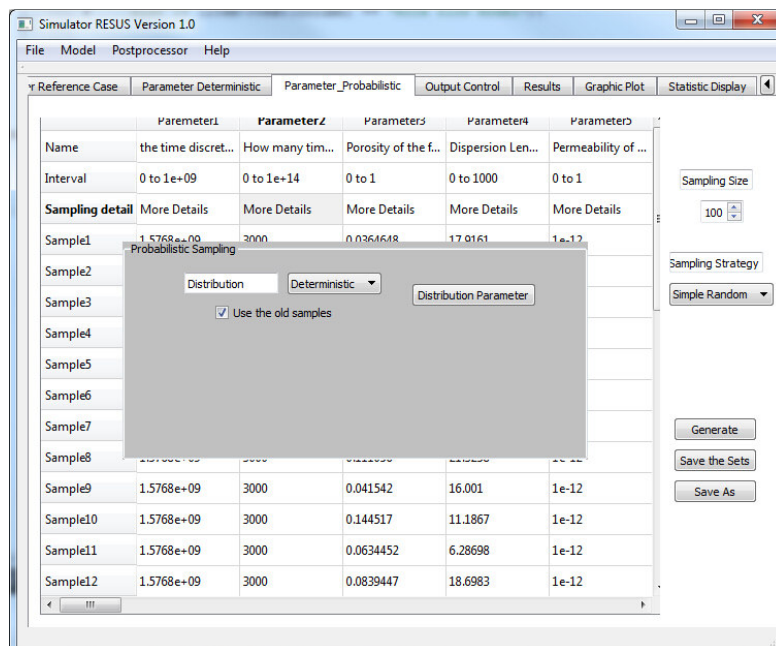


Abb. 4.45 Option „Deterministic“ als Verteilung und das „Use the old samples“ Häkchen

Die in unserem Asse-Beispiel verwendeten Verteilungen und Verteilungsmaße der probabilistischen Parameter sind in der folgenden Tabelle aufgeführt. Die Verteilungen mit ihren Verteilungsmaßen werden anhand von Messdaten und „Expert Judgement“ bestimmt [60].

Tabelle 4-5 Verteilungen mit Verteilungsmaßen für die probabilistischen Parameter

Parameter	Verteilung	Verteilungsmaße
Porosität	Gleichverteilung	0,03 – 0,15 [-]
Dispersionslänge	Dreieckverteilung	5,0 - 20,0 - 31,0 m
Speicherkoeffizient	Deterministisch	0,0 [-]
Permeabilität	Deterministisch	$10^{-12} \text{m}^2$
Gesteinsdichte	Deterministisch	27000 kg/m <sup>3</sup>
Formationslänge	Gleichverteilung	300 – 420 m
Diffusionskoeffizient	Deterministisch	$10^{-5} \text{m}^2/\text{s}$
Freisetzungskonzentration von Pu-239	Deterministisch	$1,52 \cdot 10^{-5} \text{kg/m}^3$
Sorptionskoeffizient $K_d$ von Pu-239	Dreieckverteilung	0,193 – 4,5 kg/kg
Freisetzungskonzentration von U-235	Deterministisch	$5,02 \cdot 10^{-4} \text{kg/m}^3$
Sorptionskoeffizient $K_d$ von U-235	Deterministisch	0,0005 kg/kg
Freisetzungskonzentration von Pa-231	Deterministisch	$1,74 \cdot 10^{-9} \text{kg/m}^3$
Sorptionskoeffizient $K_d$ von Pa-231	Dreieckverteilung	1,3 – 4,0 – 7,4 kg/kg
Volumenstrom	Dreieckverteilung	0,2 – 0,5 – 1,0 m/a
Fluiddichte	Deterministisch	1300 kg/m <sup>3</sup>
Fluidviskosität	Deterministisch	0,001 Pa×s
Druck am Anfang des Transportpfads	Deterministisch	0 Pa
Druck am Ende des Transportpfads	Deterministisch	20000 Pa
Zeitpunkt des Freisetzungsanfangs	Deterministisch	0,0 a
Zeitpunkt des Freisetzungsendes	Deterministisch	0,0 a

Parameter	Verteilung	Verteilungsmaße
Gesamtvolumen der Lösung	Deterministisch	$1,0 \cdot 10^6 \text{ m}^3$
Zeitschritt der Simulation	Deterministisch	$1,57 \cdot 10^{-9} [-]$
Anzahl der Zeitschritte der Simulation	Deterministisch	3000 [-]
Länge der Geometrieelemente (FEM)	Deterministisch	1,0 [-]
Zerfallskonstante von Pu-239	Deterministisch	$9,12 \cdot 10^{-13} [-]$
Zerfallskonstante von U-235	Deterministisch	$3,14 \cdot 10^{-17} [-]$
Zerfallskonstante von Pa-231	Deterministisch	$6,67 \cdot 10^{-13} [-]$
Maximale Anzahl der Iterationen für die Druckrechnung	Deterministisch	1000 [-]
Toleranz der Iteration für Druckrechnung	Deterministisch	$1,52 \cdot 10^{-5} [-]$
Maximale Anzahl der Iteration für die Berechnung der Stoffkonzentration	Deterministisch	1000 [-]
Toleranz von Iteration für die Berechnung der Stoffkonzentration	Deterministisch	$10^{-12} [-]$

Wie oben erläutert, werden die Werte der Parameter „Druck am Anfang des Transportpfads“ und „Zeitpunkt des Freisetzungsendes“ für die Simulation nicht verwendet, da bereits die Parameter „Gesamtvolumen der Lösung“ und „Freisetzung-Volumenstrom“ vorgegeben sind.

Die Option „Use the old samples“ (siehe Abb. 4.45) wird für spezifischen Modellfälle geplant. Z.B. müssen bei diesem Analysebeispiel die zu betrachtenden Radionuklide in vier Gruppen verteilt und separat berechnet werden, da die Einstellung der Simulationszeitschritte und die Modelleinstellungen für die Zerfallskette anders als für Radionuklide ohne Zerfallskette gesetzt werden müssen. Andererseits muss aber die betrachtete geologische Formation für jede Realisierung identisch für alle vier Gruppen sein, also müssen die Samplewerte für die geologischen Parameter auch identisch sein. Aus diesem Grund bietet RESUS die folgende Möglichkeit: Wenn das Modell nicht neu erstellt, sondern von einer Modelldatei von RESUS gelesen wird, können manche gewählten Parameter die Samples vom alten Modell behalten, und die anderen Parameter können neu gesampelt werden. Aber diese Möglichkeit wird nur für die spezifischen Fälle geplant, das Einlesen und Speicherung eines RESUS-Modells wird in Kapitel 4.6 zusammengefasst.

In unserem Beispiel werden die Werte beim Drücken des „Generate“ Knopfs gesampelt, wir legen die Anzahl der Samples auf 100 fest.



Nach der Samplegenerierung springt RESUS zum nächsten Tab „Output Control“, wo die Ausgabeparameter ausgewählt werden (Siehe Abb. 4.46). „Nuclid concentration 1, 2, 3“ stehen für die Konzentrationsdaten Pu-239, U-235 und Pa-231 in der Zerfallskette. Bei diesem Beispiel soll die Ausgabe der Konzentrationen der drei betrachteten Radionuklide am Endpunkt der betrachteten Geoformation erfolgen. Die drei Parameter werden in der „Output“ Spalte auf „y“ gesetzt.

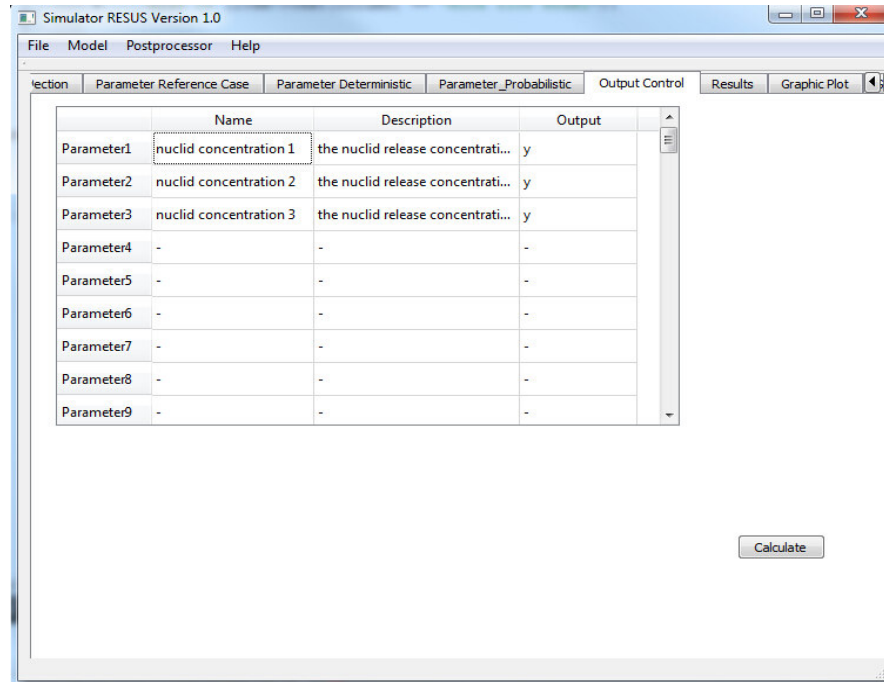


Abb. 4.46 Tab für die Ausgabekontrolle

Nach der Bestätigung mit Klick auf „Calculate“ ruft RESUS das externe RockFlow-Programm auf und lässt dort die 100 Simulationsrechnungen durchführen. Die Ergebnissedaten werden im Anschluss von RESUS zur weiteren Darstellung und Analyse eingelesen.

Im Folgenden werden die Ergebnisse für die drei Konzentrationen der betrachteten Radionuklide aus dem Asse-Transportmodell diskutiert.

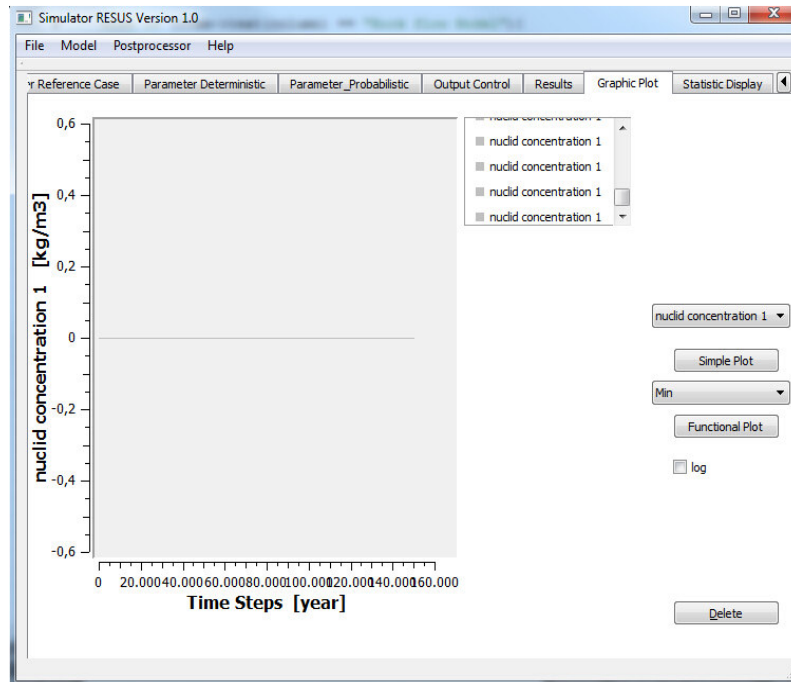


Abb. 4.47 Konzentration von Pu-239 am Ende des Transportpfads

In Abb. 4.47 ist der Konzentrationsverlauf über den Zeitraum von 60000 Jahren von Pu-239 konstant bei 0, es wird also keine Konzentration errechnet. Das liegt an der starken Adsorption von Pu-239 durch den Transportpfad (Der  $K_d$ -Wert beträgt 0,193 bis 4,5 kg/kg).

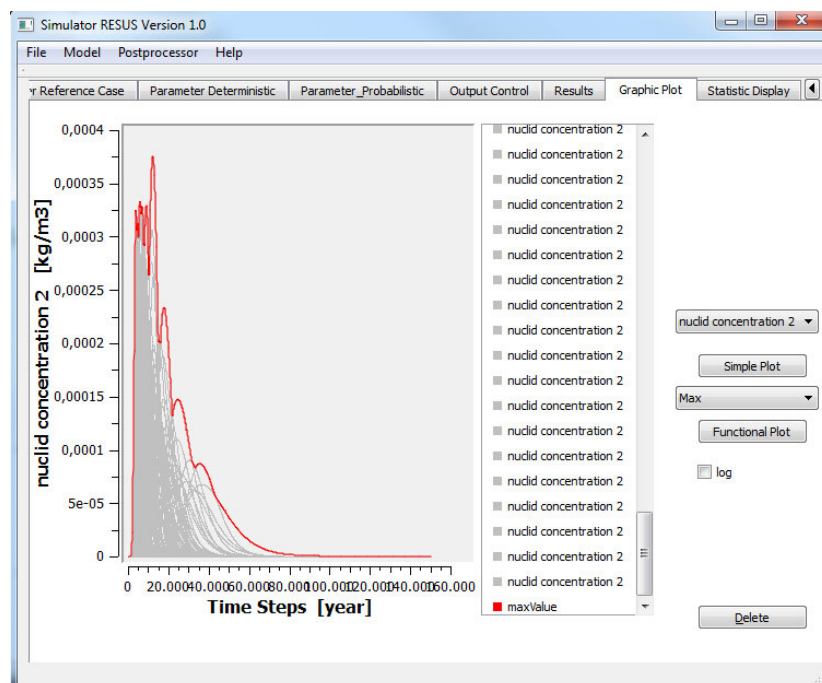


Abb. 4.48 Konzentration von U-235 am Ende des Transportpfads. Die rote Kurve markiert die maximalen Werte aller Realisierungen

Bei Abb. 4.48 treten die maximalen Konzentrationswerte im Zeitraum zwischen 2500 Jahren bis 20000 Jahren auf. Der  $K_d$ -Wert von U-235 ist deutlich kleiner im Vergleich zu Pu-239, folglich ist die Transportverzögerung durch Adsorption hier deutlich geringer ( $K_d = 0,0005$  kg/kg). Die größte Konzentration wird mit etwa  $0,000375$  kg/m<sup>3</sup> erreicht.

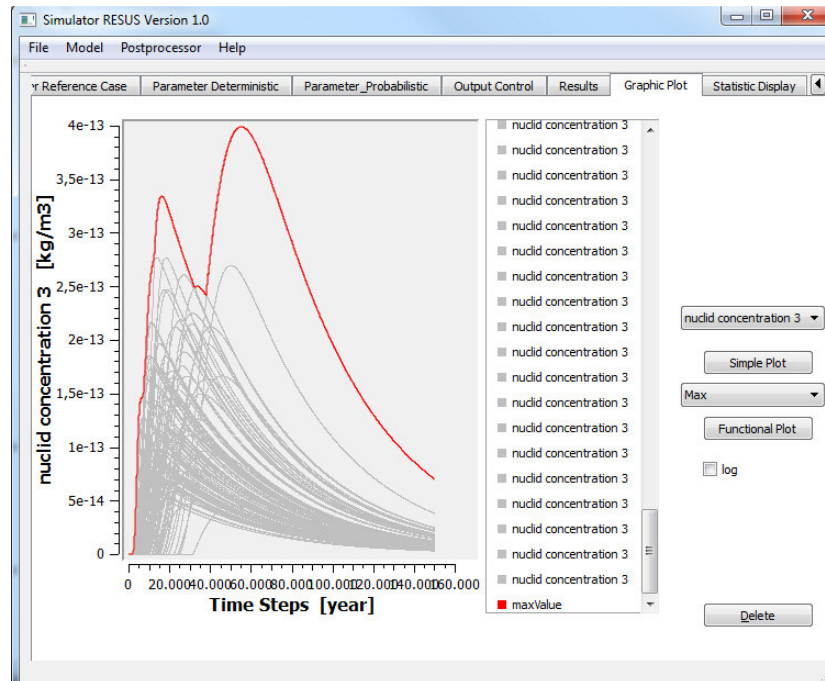


Abb. 4.49 Konzentration von Pa-231 am Ende des Transportpfads. Die rote Kurve markiert die maximalen Werte aller Realisierungen

In Abb. 4.49 ist zu sehen, dass die Konzentration des Tochternuklids Pa-231 wie erwartet deutlich geringer ausfällt als die des Mutternuklids U-235. Pa-231 hat in diesem Modell von Anfang an die Konzentration  $1,74 \cdot 10^{-9}$  kg/m<sup>3</sup> und wird mit dem  $K_d$ -Wert  $1,3 - 4,0 - 7,4$  kg/kg absorbiert. Die Peaks kommen im Zeitraum von 5000 Jahren bis 80000 Jahren vor, etwas später als die Peaks von U-235. Dies passt gut zu der Reihenfolge der Zerfallskette. Der maximale Wert aller Realisierungen beträgt  $4 \cdot 10^{-13}$  kg/m<sup>3</sup>.

Nach den allgemeinen Darstellungen der Ergebnisse können die Ergebnisdaten im nächsten Tab statistisch weiter analysiert werden.

Die Histogrammdarstellung wird im Folgenden nur für die Nuklide U-235 und Pa-231 durchgeführt, da für Pu-239 keine Konzentration errechnet wurde. Der angewendete „Quantity of interest“ ist „Max per realization“ (siehe Abb. 4.50)

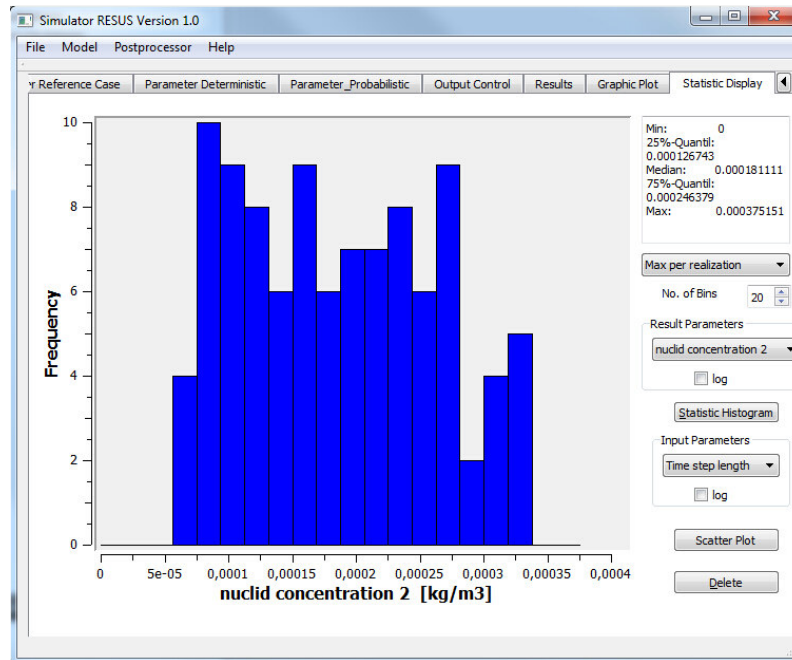


Abb. 4.50 Histogramm der maximalen Werte der Konzentration von U-235 mit 20 Wertklassen. Die Quantile sind im Textfeld rechts oben abzulesen

Die Balken des Histogramms der maximalen Werte der Konzentration von Pa-231 in Abb. 4.51 liegen zwischen den Konzentrationen von  $4,0 \cdot 10^{-14}$  und  $3,4 \cdot 10^{-13}$  kg/m<sup>3</sup>. Die Verteilung kann nicht zu einem bekannten Verteilungstyp zugeordnet werden.

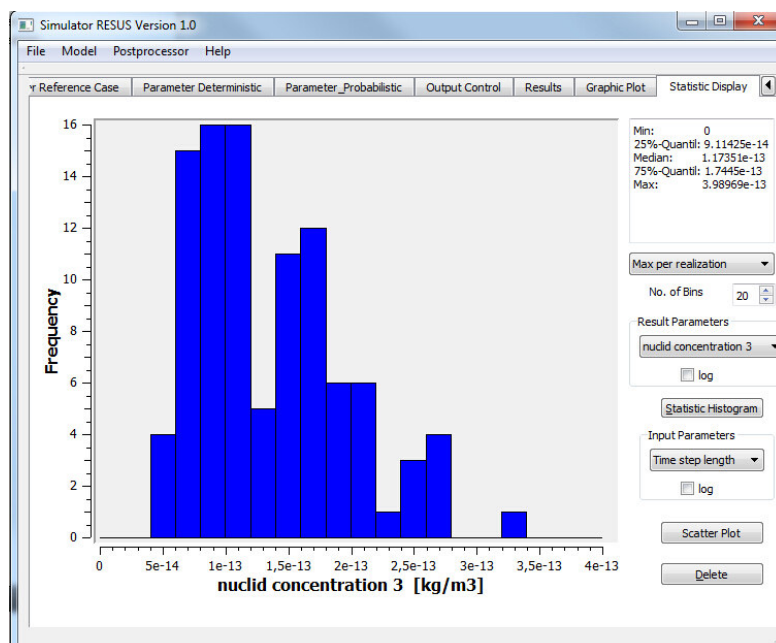


Abb. 4.51 Histogramm der maximalen Werte der Konzentration von Pa-231 mit 20 Wertklassen

Nach der Histogrammanalyse werden Sensitivitätsanalysen mit Hilfe der Streudiagrammdarstellung (Scatterplot) durchgeführt.

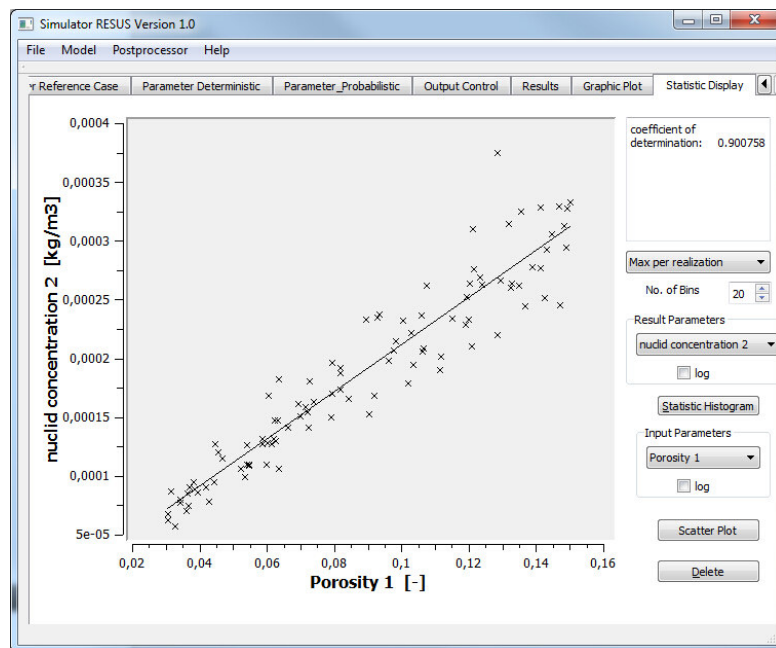


Abb. 4.52 Streudiagramm: Probabilistische Porositätswerte des Transportpfads gegen die maximalen Konzentrationen aller Rechenrealisierungen von U-235

Abb. 4.52 zeigt, dass die Konzentration von U-235 mit der Porosität des Mediums stark korreliert, mit Bestimmtheitsmaß gleich 0,9. Dafür könnte es zwei verschiedene Ursachen geben, die kurz dargelegt werden sollen:

1. U-235 Nuklid wird durch die Grundwasserströmung transportiert. Allgemein besteht ein physikalischer Zusammenhang zwischen der Porosität eines Mediums und der Abstandsgeschwindigkeit einer durchfließenden Wasserströmung. Je größer die Porosität ist, desto größer ist die Abstandsgeschwindigkeit. Deswegen ist Porosität ein positiver Faktor für den Stofftransport.
2. U-235 hat in diesem Beispiel einen konstanten  $K_d$  Wert. Die Porosität der Geoformation spielt eine Rolle beim Adsorptionsprozess. Letzterer nimmt wiederum Einfluss auf die Konzentration von U-235. Der Retardationsfaktor wird wie folgt berechnet:

$$R_d = 1 + \frac{K_d \cdot \delta_{tr}}{n} = \frac{V_{wasser}}{V_{Stoff}} \quad (4-12)$$

Dabei ist  $\delta_{tr}$  die Dichte des betrachteten Gesteins [g/cm<sup>3</sup>],  $n$  ist die Porosität [-],  $K_d$  ist der Adsorptionskoeffizient [-],  $V_{wasser}$  ist die Fließgeschwindigkeit der Wasserströmung [m/a],  $V_{Stoff}$  ist die Transportgeschwindigkeit [m/a].

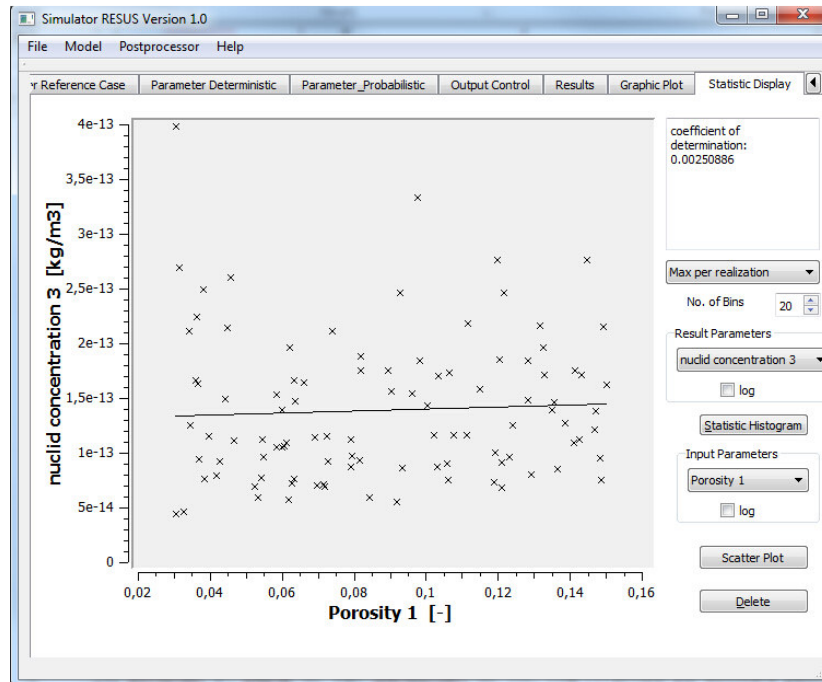


Abb. 4.53 Streudiagramm (Scatterplot): probabilistische Porositätswerte gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfads

Wie in Abb. 4.53 zu erkennen ist, besteht zwischen der maximalen Konzentration von Pa-231 und der Porosität keine lineare Korrelation, im Gegensatz zum Ergebnis für U-235. Dies liegt in erster Linie an der probabilistischen Berechnung des  $K_d$ -Werts für Pa-231. Dieser nimmt nach Gleichung 4-12 starken Einfluss auf die Konzentration von Pa-231 und überlagert auf diese Weise den Einfluss der Porosität. Dies wird durch Abb. 4.54 bestätigt. Sie zeigt, dass der  $K_d$ -Wert einen starken negativen Einfluss auf den Transport von Pa-231 hat, was den Einfluss aus der Porositätsänderung des Transportpfads überlagert.

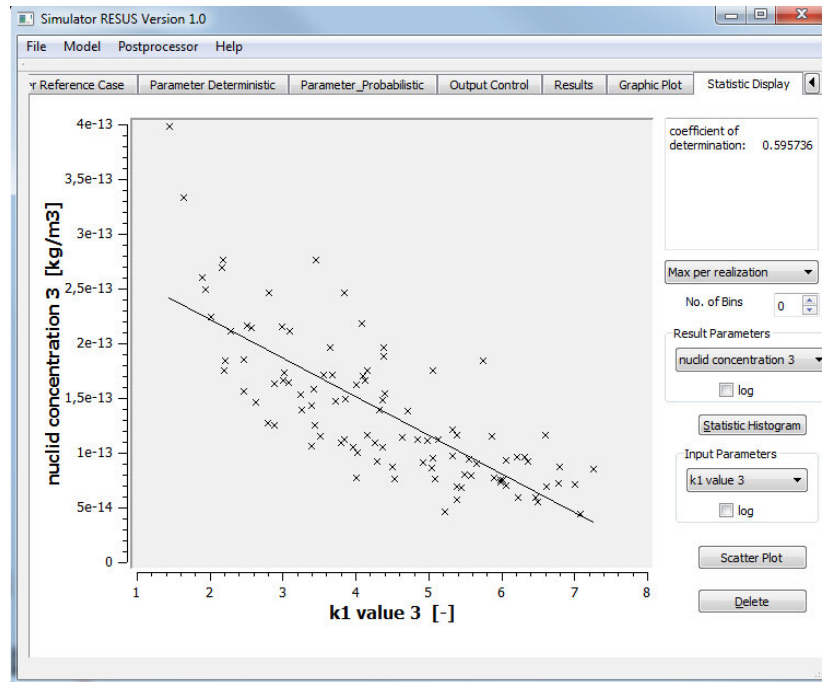


Abb. 4.54 Streudiagramm (Scatterplot): probabilistische  $K_d$ -Werte gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfads

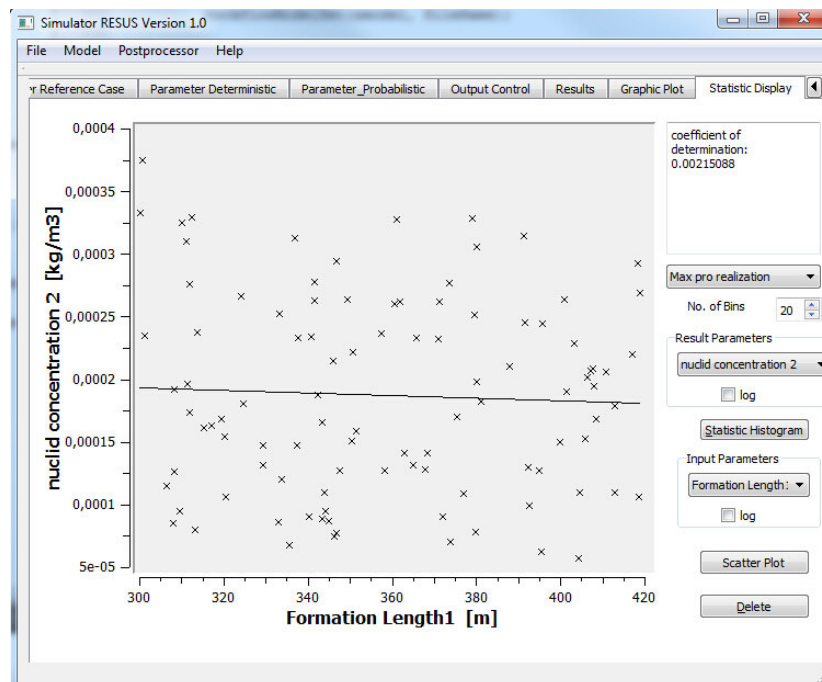


Abb. 4.55 Streudiagramm (Scatterplot): probabilistische Weglängen des Transportpfades gegen die maximalen Konzentrationen von U-235 am Ende des Transportpfades



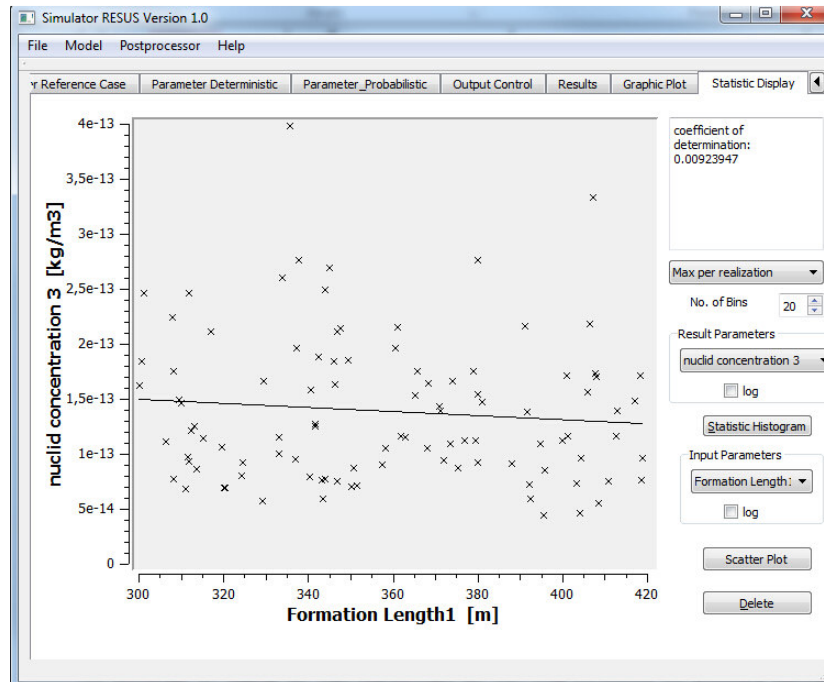


Abb. 4.56 Streudiagramm (Scatterplot): probabilistische Weglängen des Transportpfades gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfades

An den Abb. 4.55 und Abb. 4.56 kann man erkennen, dass die betrachteten Radionuklidkonzentrationen keine eindeutige lineare Beziehung zu den Weglängen der Transportpfade haben. Aus den Sensitivitätsanalysen für dieses advektionsdominierte Transportbeispiel kann man die Schlussfolgerung ziehen, dass die Adsorptionsfaktoren und die daraus hergeleiteten Retardationsfaktoren für die stark absorbierten Radionuklide (wie z.B. U-235 und Pa-231) entscheidende Eingabeparameter sind. Der Einfluss der Parameter Pfadlänge und Volumenstrom auf die Konzentration der Nuklide ist deutlich geringer als der Adsorptionseffekt.



## 4.4 Zweidimensionales Transport-Modell in RESUS

### 4.4.1 Allgemeine Vorstellung der zweidimensionalen Transport-Modelle

Als Erweiterung des eindimensionalen Transportmodells wurde ein zweidimensionales Transportmodell in RESUS implementiert, welches auch mit der GUI-basierten Methode konfiguriert werden kann. Das zweidimensionale Transportmodell beinhaltet nicht nur die geometrische Erweiterung des eindimensionalen Modells, sondern berücksichtigt zusätzlich weitere Transporteffekte, wie z.B. Geoanomalien. Die Nutzung dieses Modells ist insbesondere für Lehre und Ausbildung von Interesse, da die hier getroffenen Vereinfachungen die tatsächlichen Bedingungen nur unzureichend wiedergeben.

Die Geometrie der geologischen Transportmedien wird durch das zweidimensionale Transportmodell in rechteckigen Flächen nachgebildet. Zulässig sind mehrere geologische Transportschichten, die überdies beliebig viele geologische Unregelmäßigkeiten beinhalten dürfen. Letztere werden hier als „Geoanomalie“ bezeichnet. Die Breite und Länge der zu simulierenden Geoformation legt die Geometrie des Modells fest.

Ein Beispielmmodell wird in Abb. 4.57 gezeigt. Das zweidimensionale Modell kann vertikal oder horizontal verwendet werden. Die Geoformationen liegen übereinander bei einem vertikalen Modell und nebeneinander bei einem horizontalen Modell. Die Geometrie der Schichten (Transportmedien) wird durch Längen- und Breitenangaben festgelegt. Die Geoanomalien werden durch die Definition der Koordinaten ihrer Anfangs- und Endpunkte gegeben. Die Koordinate an der linken unteren Ecke ist der Nullpunkt (0, 0).

Ebenso werden durch Koordinatenangabe ein Freisetzungspunkt und ein Messpunkt positioniert. An dem Ersteren tritt das betrachtete Radionuklid in das Modell ein, der Letztere dient zur Festlegung der Messortes der Nuklidkonzentration.

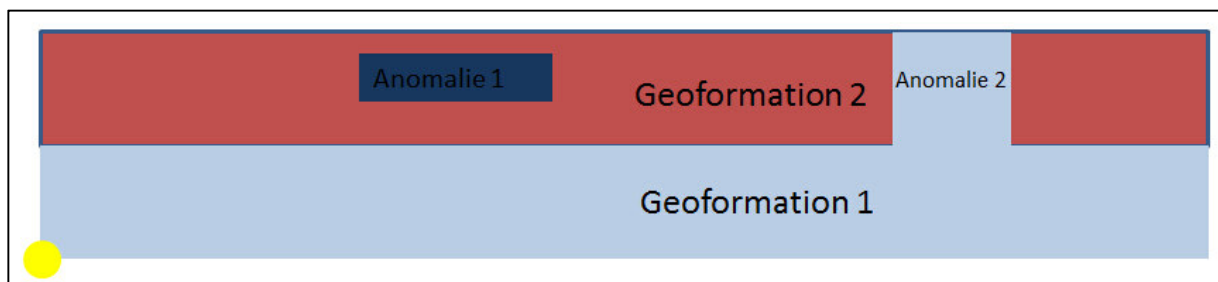


Abb. 4.57 Beispiel eines zweidimensionalen Transportmodells mit zwei geologischen Schichten und zwei Geoanomalien

In Abb. 4.57 gibt es zwei Geoanomalien in der Geoschicht 1. Die dunkelblaue Anomalie 1 ist in Geoschicht 1 eingeschlossen. Die hellblaue Anomalie 2 hat die gleichen Materialeigenschaften wie Geoschicht 2 und durchdringt die Geoschicht 1.

#### 4.4.2 Ein Beispiel für ein zweidimensionales Transportmodell in RESUS

In diesem Kapitel wird Schritt für Schritt beschrieben, wie ein zweidimensionales Transportmodell in RESUS erstellt wird. Dies geschieht anhand eines einfachen Beispielsmodells, das lediglich zu Demonstrationszwecken dient.

Das Beispielsmodell besteht aus einer Geoschicht mit einer Geoanomalie (siehe Abb. 4.58).

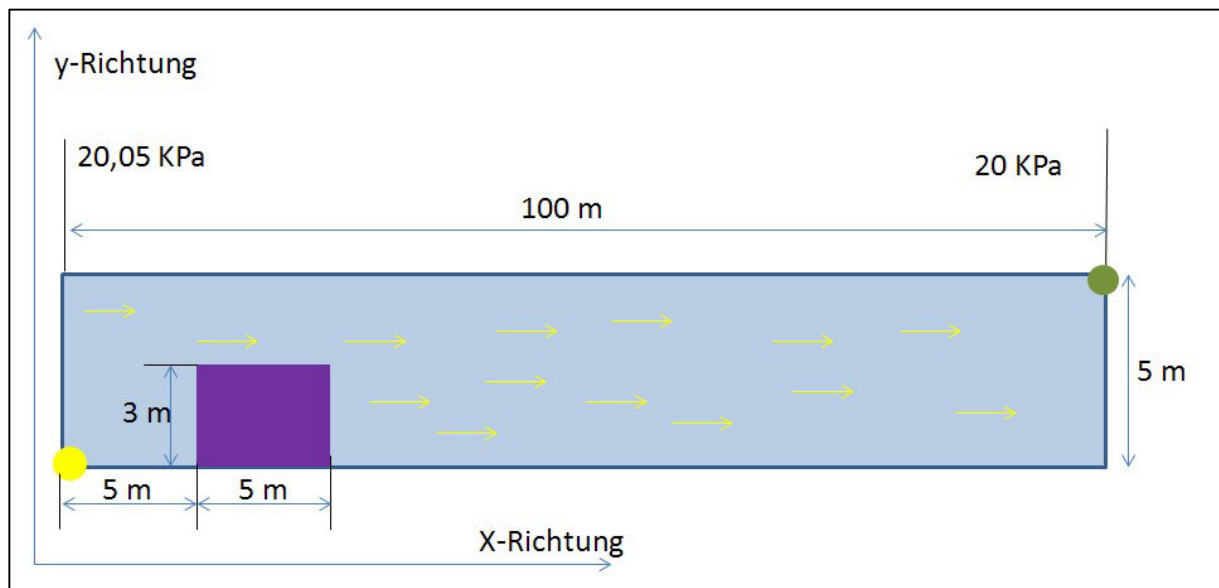


Abb. 4.58 Zweidimensionales Beispielsmodell zur Demonstration in RESUS

Zunächst sollen die in Abb. 4.58 dargestellten Eigenschaften des Beispielsmodells verdeutlicht werden. Das hellblaue Gebiet ist der Grundwasserleiter mit 100 m Länge und 5 m Höhe. Auf der linken Eintrittsseite herrscht ein Wasserdruck von 20,05 kPa und auf der rechten Austrittsseite 20 kPa. Es besteht folglich ein Wasserdruckgradient, die Fließrichtung ist entsprechend von links nach rechts. Der gelbe Punkt ist die Freisetzungsquelle der Schadstoffe. Für diese Stelle gilt die später anzugebende Randbedingung für die Nuklidkonzentration. Der grüne Punkt ist der Messpunkt, an dem die Nuklidkonzentration beobachtet wird. Der violette Bereich stellt die Geoanomalie dar, die 5 m in der Länge und 3 m in der Höhe misst. Die Geoanomalie befindet sich im linken Bereich des Modells, mit einem Abstand von 5 m zum linken Modellrand.

Wie dargestellt, wird das aus dem Freisetzungspunkt ausgetretene Radionuklid von der Grundwasserströmung von links nach rechts transportiert. Allgemein wird die Geschwindigkeit der Grundwasserströmung entweder durch den Wasserdruckgradienten oder den Wasserzufluss bestimmt. Um die Modellrechnung zu vereinfachen, wird in unserem Beispiel nur ein Radionuklid betrachtet. Die Geoanomalie hat eine niedrigere Porosität und Permeabilität als

der Grundwasserleiter. In der longitudinalen Richtung (X-Richtung) wird das Nuklid hauptsächlich advektiv transportiert. Die Ausbreitung in transversaler Richtung (Y-Richtung) ist durch den diffusiven Transport dominiert. Durch die Behinderung von der Geoanomalie mit niedrigerer Permeabilität kann der vertikale Transport auch vor dem Anomaliebereich stattfinden. Die Ermittlung der Konzentration des Nuklids am Messpunkt ist das Ziel der Modellierung.

Die in RESUS festzulegenden Parametereinstellungen zur Nachbildung dieses Modells werden im Zuge der allgemeinen Beschreibung der Modellerstellung angegeben.

Um in RESUS ein zweidimensionales Transportmodell zu erstellen, ist zunächst wieder die Modellauswahl in „Model Selection“-Tab nötig. Dort erscheint unter dem Punkt „Geosphere Transport“ die hier benötigte Option „2-D\_RockFlow“. Ebenso wie beim eindimensionalen Transportmodell kann man bei Menü unterscheiden, mit welchem Programm das Modell berechnet werden soll. Da für dieses Modell beider Programme (RockFlow und OGS) die gleichen Ergebnisse liefern, zeigen wir in dieser Dissertation das Transportmodell, das mit RockFlow berechnet wird. Durch einen Doppelklick auf „2-D\_RockFlow“ erscheint ein zusätzliches Fenster für die detaillierte Modelleinstellung. Darin muss die Anzahl der betrachteten Geoformationen, Radionuklide, Geoanomalien und die Adsorptionsisotherme festgelegt werden (siehe Abb. 4.59). In unserem Beispiel sind alle Werte mit „1“ und das Adsorptionskonzept mit „Henry (linear)“ zu wählen. Der in unserem Modell beobachtete Schadstoff ist das Radionuklid U-235.

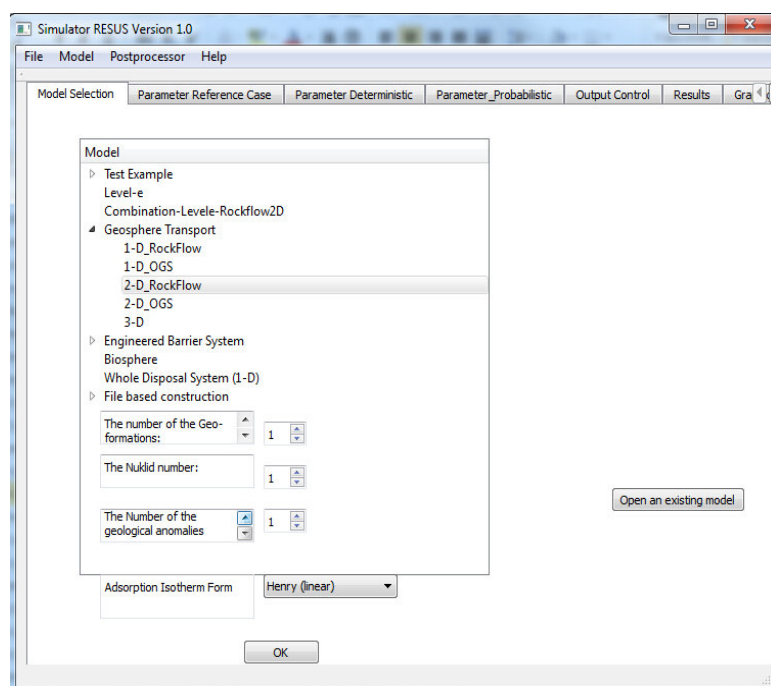


Abb. 4.59 Modellauswahl- und Einstellung im ersten Tab für das zweidimensionale Transportmodell

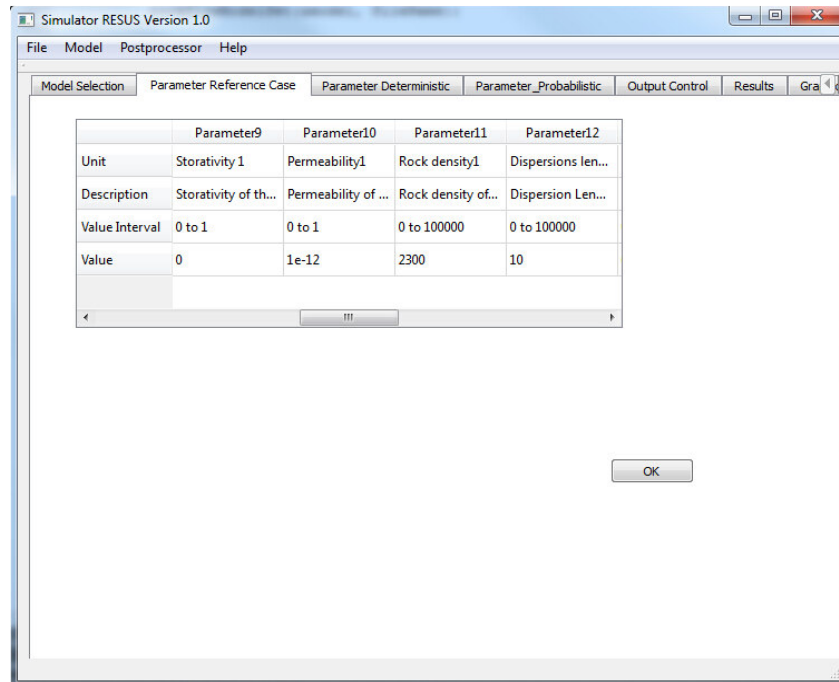


Abb.4.60 „Parameter Reference Case“-Tab zur Angabe der Referenzwerte der probabilistischen Parameter

Nach Angabe der Modelleinstellungen im „Model Selection“-Tab und Bestätigung der Eingaben mit „OK“ geht RESUS zum nächsten „Parameter Reference Case“-Tab über. Darin werden für die probabilistischen Parameter Referenzwerte angegeben. Dies ist wie schon bei den vorangegangenen Beispielen nötig, um einen Testdurchlauf mit bekannten Parametern machen zu können. Die in „Parameter Reference Case“ Tab für unser zweidimensionales Modell anzugebenden Werte sind in der folgenden Tabelle aufgeführt:

Tabelle 4-6 Einstellung der Referenzwerte für die probabilistischen Parameter

Parametername	Einheit	Referenzwert
X-Koordinate vom Freisetzungspunkt	$m$	0
Y-Koordinate vom Freisetzungspunkt	$m$	0
Wasserzufluss	$m^3/s$	0,0
Dichte der Flüssigkeit	$kg/m^3$	1300
Viskosität der Flüssigkeit	$Pa \cdot s$	0,001
Wasserdruck auf der linken Modellseite	$Pa$	20500
Wasserdruck auf der rechten Modellseite	$Pa$	20000

Parametername	Einheit	Referenzwert
Porosität der betrachteten Geoformation	-	0,2
Speicherkoeffizient der Geoformation	-	0,0
Permeabilität der Geoformation	$m^2$	$1 \cdot 10^{-12}$
Dichte der Geoformation	$kg/m^3$	2300
Dispersionslänge der Geoformation	$m$	10
Porosität der Geoanomalie	-	0.15
Speicherkoeffizient der Geoanomalie	-	0,0
Permeabilität der Geoanomalie	$m^2$	$1 \cdot 10^{-13}$
Dichte der Geoanomalie	$kg/m^3$	2500
Dispersionslänge der Geoanomalie	$m$	10,0
Diffusionskoeffizient der Lösung	$m^2/s$	$1 \cdot 10^{-5}$
$K_d$ Wert für das Nuklid	$kg/kg$	0,005
Freisetzungskonzentration	$kg/m^3$	1,0

Die angegebenen X- und Y-Koordinaten legen die Position des Freisetzungspunkts fest. Bei diesem Beispiel wird der Wasserzufluss auf 0 gesetzt, die Wasserströmung ist hier also durch die Wasserdruckdifferenz determiniert, der 500 Pa beträgt. Weiterhin werden die Materialparameter der Geoformation, Geoanomalie und des Grundwasserseingestellt.  $K_d$  wird für den Testdurchlauf zunächst auf 0,005 kg/kg gesetzt, in den weiteren Realisierungen bleibt es in der Größenordnung von  $10^{-3}$  kg/kg. Damit soll sichergestellt werden, dass das betrachtete Radionuklid nur schwach absorbiert wird. Auf diese Weise stehen für die Analyse am Ende mehr Konzentrationsdaten zur Auswertung zur Verfügung, was in Lehre und Ausbildung didaktisch günstig ist (siehe Abb. 4.58).

Nach der Einstellung der Referenzwerte der probabilistischen Parameter geht die GUI von RESUS zum nächsten Tab „Parameter Deterministic“ über. Dieser ist für die Einstellung der deterministischen Parameter der Modellrechnung gedacht (siehe Abb. 4.61). Diese werden als Konstanten in allen Rechenrealisierungen eingesetzt.

Wie schon bei den probabilistischen Parametern hängt die Anzahl der deterministischen Parameter von der gewählten Anzahl der Geoformationen, Radionuklide und Geoanomalien ab. Bei diesem Beispiel gibt es 21 deterministische Parameter, die in Tabelle 4-7 dargestellt werden.

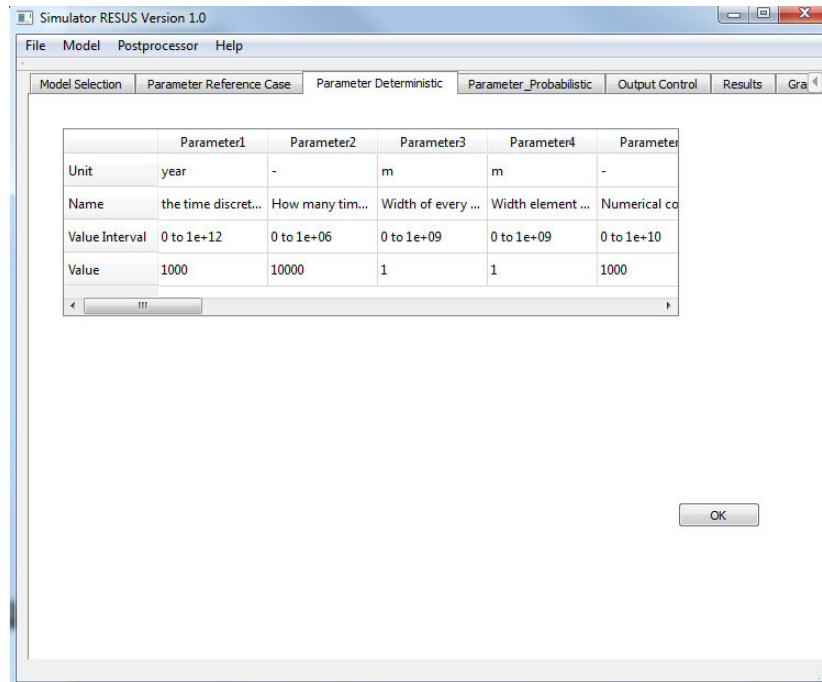


Abb. 4.61 Tab „Parameter Deterministic“ für die Werteinstellung der deterministischen Parameter

Tabelle 4-7 Deterministische Werteinstellung der Parameter

Parametername	Einheit	Wert
Zeitschrittlänge der Simulation	<i>s</i>	86400
Anzahl der zu simulierenden Zeitschritte	-	1000
Die Breite des Modells (Y-Richtung)	<i>m</i>	5
Länge eines FEM-Elements in Y-Richtung	<i>m</i>	1
Maximale Anzahl von FEM-Iterationen für die Wasserströmungsrechnung	-	1000
Numerische Toleranz für die Iteration der Wasserströmungsrechnung	-	$1 \cdot 10^{-12}$
Maximale Anzahl von FEM-Iterationen für die Stofftransportrechnung	-	1000
Numerische Toleranz für die Iteration der Stofftransportrechnung	-	$1 \cdot 10^{-12}$
Länge der Geoformation	<i>m</i>	100

Parametername	Einheit	Wert
Länge eines FEM-Elements in X-Richtung	$m$	1
Zerfallskonstante des Radionuklids	$s^{-1}$	$3,14 \cdot 10^{-17}$
minX-Wert der betrachteten Geoanomalie	$m$	5,0
maxX-Wert der betrachteten Geoanomalie	$m$	10,0
minY-Wert der betrachteten Geoanomalie	$m$	0,0
maxY-Wert der betrachteten Geoanomalie	$m$	3,0
X-Koordinate des Messpunkts	$m$	100,0
Y-Koordinate des Messpunkts	$m$	5,0

Die gesamte Simulation wird für einen Zeitraum von 1000 Tagen durchgeführt, dabei beträgt ein Zeitschritt in der Simulationsrechnung einen Tag. Die Angaben „Länge/Breite eines FEM-Elements“ beschreiben die Gittergröße im Finite-Elemente-Modell. Die hier gemeinte Breite kann im zweidimensionalen Modell die vertikale Mächtigkeit oder die horizontale Breite der Geoformation bezeichnen. (Die Beschreibung der Methodik für den Modellaufbau ist in Kapitel 5.2.2 zu finden). Die Koordinaten der Geoanomalie (minX, maxX, minY, maxY) legen die Position und Größe der rechteckigen Geoanomalie fest. Die Koordinaten des Messpunkts positionieren den Punkt, an dem die berechnete Stoffkonzentration gemessen wird. Das betrachtete Radionuklid U-235 hat eine Halbwertszeit von  $7,0 \cdot 10^8$  Jahren, was einer Zerfallskonstante von  $3,14 \cdot 10^{-17} s^{-1}$  entspricht. Die Halbwertszeit von U-235 ist im Verhältnis zur Simulationszeit dieses Beispiels sehr groß, so dass der Zerfall des Nuklids hier fast keine Rolle spielt.

Der nächste Schritt im Pre-Processing-Vorgang ist die Samplegenerierung. Sie findet in Tab „Parameter\_Probabilistic“ statt. Unser Beispielmmodell verwendet 20 probabilistische Parameter. Diese sind in der Tabelle zunächst mit der Default-Verteilung „Deterministic“ aufgelistet. Nur für einige Parameter sind in diesem Beispiel bestimmte statistische Verteilungen anzugeben. Für die restlichen probabilistischen werden die deterministischen Referenzwerte von aus der Tabelle 4-6 in der Berechnung verwendet.

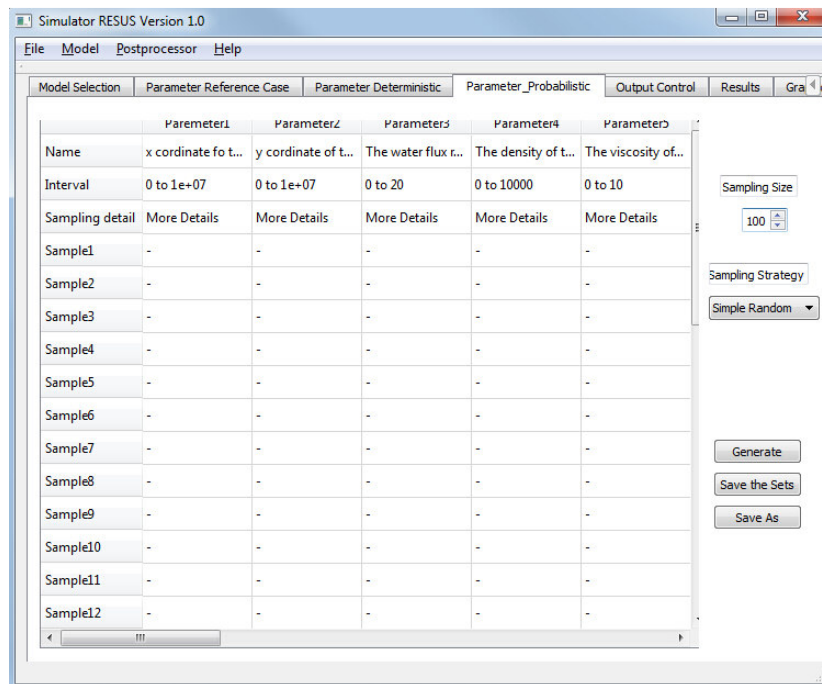


Abb. 4.62 Tab „Parameter\_Probabilistic“ für die Einstellung der Verteilungen der Parameter

Die Anzahl der Zufallsgenerierungen (Realisierungen) wird auf 100 gesetzt. Alle Parameter aus unserem Beispiel, für die bestimmten Verteilungen festgelegt werden müssen, sind mit diesen Verteilungen und deren Verteilungsmaßen in der folgenden Tabelle eingetragen:

Tabelle 4-8 Verteilungseinstellung für die probabilistischen Parameter

Parametername	Statistische Verteilung	Verteilungsmaße (min-max)
Porosität der Geoformation	Gleichverteilung	0,15 – 0,35
Permeabilität der Geoformation	Log-Gleichverteilung	$1 \cdot 10^{-12}$ – $1 \cdot 10^{-11}$
Porosität der Geoanomalie	Gleichverteilung	0,1 - 0,2
Permeabilität der Geoanomalie	Log-Gleichverteilung	$1 \cdot 10^{-14}$ – $1 \cdot 10^{-12}$
Diffusionskoeffizient der Lösung	Gleichverteilung	$1 \cdot 10^{-6}$ – $1 \cdot 10^{-5}$
K <sub>d</sub> Wert von beiden Medien	Gleichverteilung	0,001 – 0,007

Die Permeabilitätswerte der zwei Transportmedien unseres Modells unterscheiden sich um mehrere Größenordnungen. Ihre Verteilung ist daher logarithmisch festgelegt, in diesem Fall ist es eine log-Gleichverteilung. Die Geoanomalie hat hier eine niedrigere Porosität und Permeabilität als die Geoformation.



Nach der Verteilungseinstellung werden durch einen Klick auf „Generate“ 100 Samples für die Parameter erzeugt. Sobald die Samples zur Verfügung stehen, öffnet sich der Tab „Output Control“. Für dieses Modellbeispiel wird nur ein Ergebnisparameter aufgelistet, da lediglich von einem einzigen Radionuklid die Konzentration gemessen wird. Der Benutzer stellt in der Ausgabeoption „y“ ein, und startet danach die Berechnung durch einen Klick auf den Knopf „Calculate“.

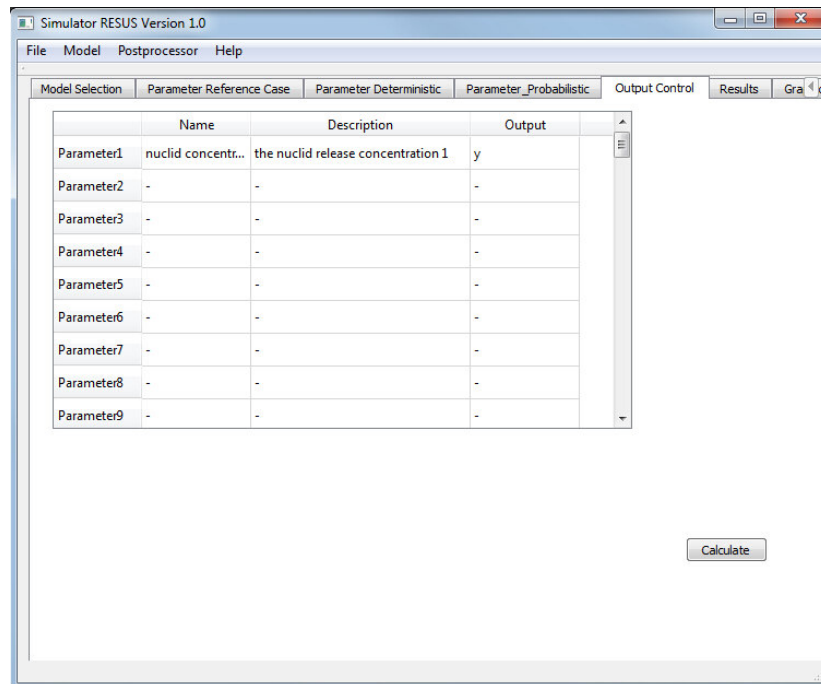


Abb. 4.63 „Output\_Control“ für den einzigen Ausgabeparameter dieses Beispiels

Die Simulation wird dann durch das RockFlow-Programm 100 Mal ausgeführt. Nach den Rechenrealisierungen kann der Post-Prozess beginnen. Zuerst wird der zeitabhängige Konzentrationsverlauf am Messpunkt dargestellt (siehe Abb. 4.64). Die statistischen Kennwerte zu jedem Zeitpunkt werden ebenfalls dargestellt.

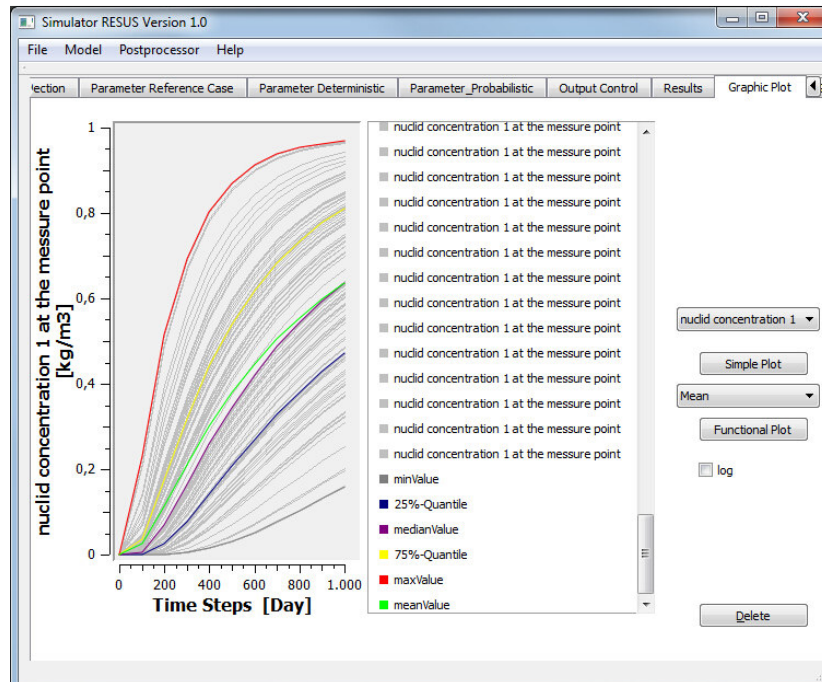


Abb. 4.64 Konzentrationsverläufe am Messpunkt von 100 Rechenrealisierungen

Anhand der Verläufe sieht man, dass die Konzentration von allen Realisierungen innerhalb der 1000 Tage verschieden stark ansteigt. Die Endwerte unterscheiden sich dementsprechend. Der maximale Wert beträgt  $0,95 \text{ kg/m}^3$ , was fast dem Freisetzungswert entspricht ( $1,0 \text{ kg/m}^3$ ).

Nach der ersten Darstellung zur Übersicht im „Graphic Plot“-Tab wechseln wir zum Tab „Statistic Display“ für die weitere Analyse. Zuerst wollen wir die Ergebnisse als Histogramm darstellen lassen (siehe Abb. 4.65). Die Analyse dieses Beispiels wird im Folgenden stets mit dem „Quantity of interest“: „Max per realization“, also dem maximalem Wert jeder Realisierung durchgeführt.

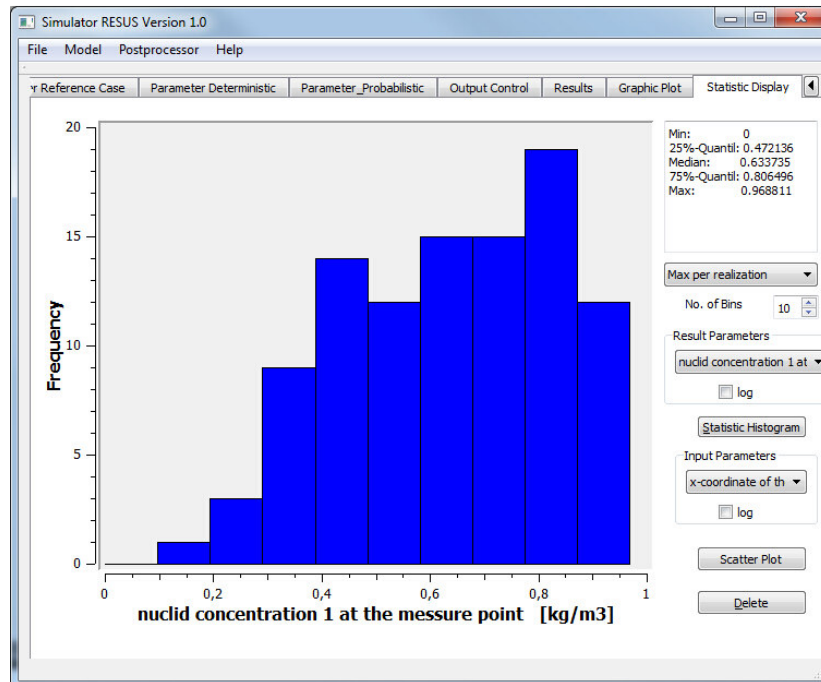


Abb. 4.65 Histogramm mit 10 Wertklassen der Konzentrationshäufigkeiten

Anhand des Histogramms ist eine rechtsschiefe Verteilung anzunehmen. Aber wegen der geringen Anzahl der Realisierungen und der dementsprechenden Breite der Wertklasse im Histogramm ist die Festlegung einer Verteilung problematisch.

Nach der Darstellung mit dem Histogramm gehen wir, wie schon in den obigen Beispielen, zur Streudiagrammdarstellung (Scatterplot) über. Diese ermöglicht die Sensitivitätsanalyse zwischen Ein- und Ausgabeparametern. Nachfolgend sollen die interessantesten Ergebnisse beispielhaft besprochen werden.

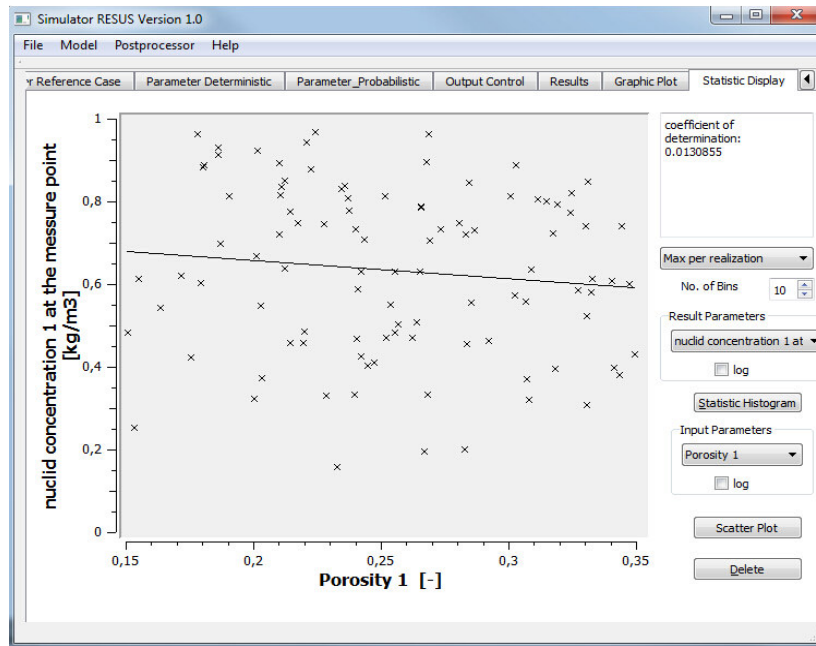


Abb. 4.66a Streudiagramm: Porosität der Geoformation [-] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m³]

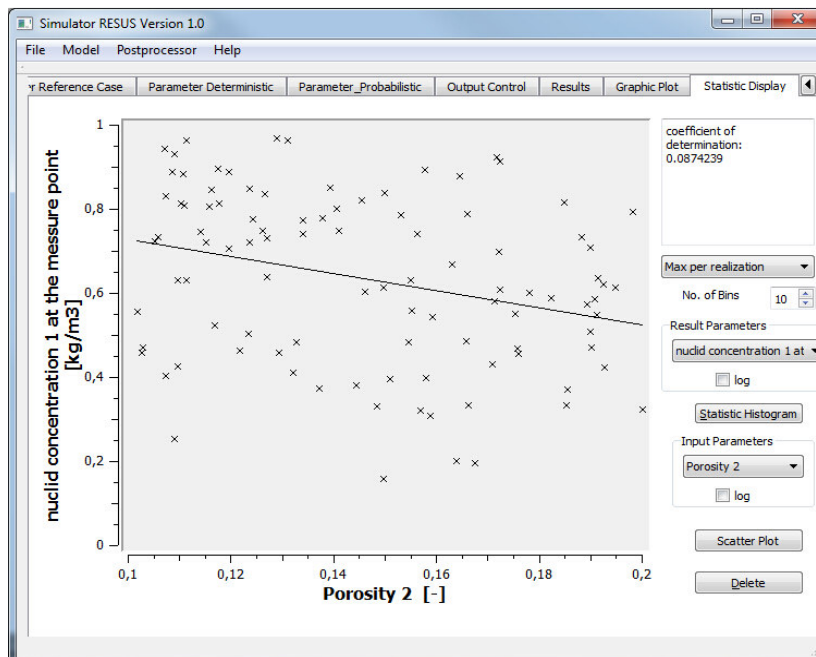


Abb. 4.66b Streudiagramm: Porosität der Geoanomalie [-] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m³]

Anhand der Abb. 4.66a und b sieht man, dass die Punktwolken sehr breit verteilt sind und die errechneten Bestimmtheitsmaße sehr niedrig sind. Es gibt also keine klare lineare Korrelation zwischen Porosität und Ergebniskonzentration. Dennoch ist die Korrelation zwischen Konzentration und Porosität der Geoanomalie ( $R^2 = 0,097$ ) etwa größer als jene mit der

Porosität der Geoformation ( $R^2 = 0,013$ ). Die in den beiden Abbildungen dargestellten Ergebnisse kann man als nicht signifikant bezeichnen. Im Vergleich zum eindimensionalen Transportmodell aus dem vorherigen Asse-Beispiel spielt die Porosität in diesem Modell eine geringere Rolle. Die Porosität geht physikalisch im eindimensionalen Modell direkt in die Berechnung der Fließgeschwindigkeit ein. Im zweidimensionalen Modell wird die Fließgeschwindigkeit mittelbar aus dem Wasserdruckgradienten und der Permeabilität nach Darcy'schem Gesetz berechnet. Die Porosität ist im zweidimensionalen Modell weniger direkt in der Berechnung der Fließgeschwindigkeit enthalten und nimmt daher weniger Einfluss auf diese und letztendlich auch auf den Schadstofftransport.

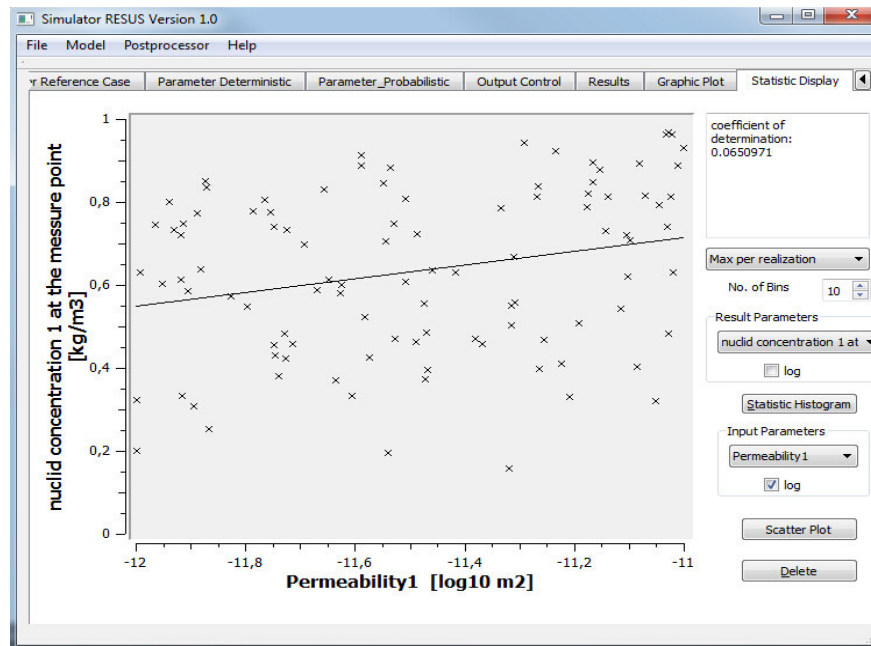


Abb. 4.67a Streudiagramm: Logarithmus der Permeabilität der Geoformation [ $\log_{10} \text{ m}^2$ ] gegen die Nuklidkonzentration von U-235 am Messpunkt [ $\text{kg}/\text{m}^3$ ]

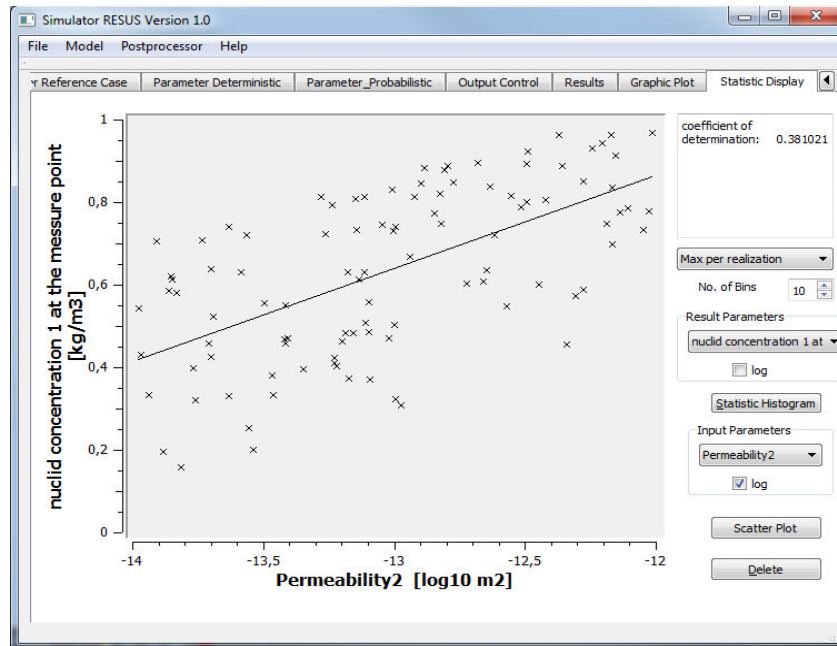


Abb. 4.67b Streudiagramm: Logarithmus der Permeabilität der Geoanomalie [ $\log_{10} \text{ m}^2$ ] gegen die Nuklidkonzentration von U-235 am Messpunkt [ $\text{kg/m}^3$ ]

Die Abb. 4.67a und b zeigen die unterschiedlichen Beziehungen zwischen der Permeabilität der Geoformation bzw. der Permeabilität der Geoanomalie mit der Ergebniskonzentration. Die logarithmische Permeabilität der Geoanomalie hat eine lineare Korrelation mit der Konzentration, mit Bestimmtheitsmaß = 0,381. Die Korrelation ist eindeutiger als die der Geoformation. Daraus kann man schließen, dass die Geoanomalie aufgrund ihrer deutlich niedrigeren Permeabilität trotz ihrer geringeren geometrischen Ausmaße beim Nuklidtransport eine wichtigere Rolle spielt als die normale Geoformation. Die Geometrie und Position der „Anomalie“ kann natürlich die Ergebnisse stark beeinflussen. In diesem Beispiel liegt die Geoanomalie sehr nah zum Freisetzungspunkt, was den Behinderungseffekt der Geoanomalie auf den Stofftransport verstärkt.

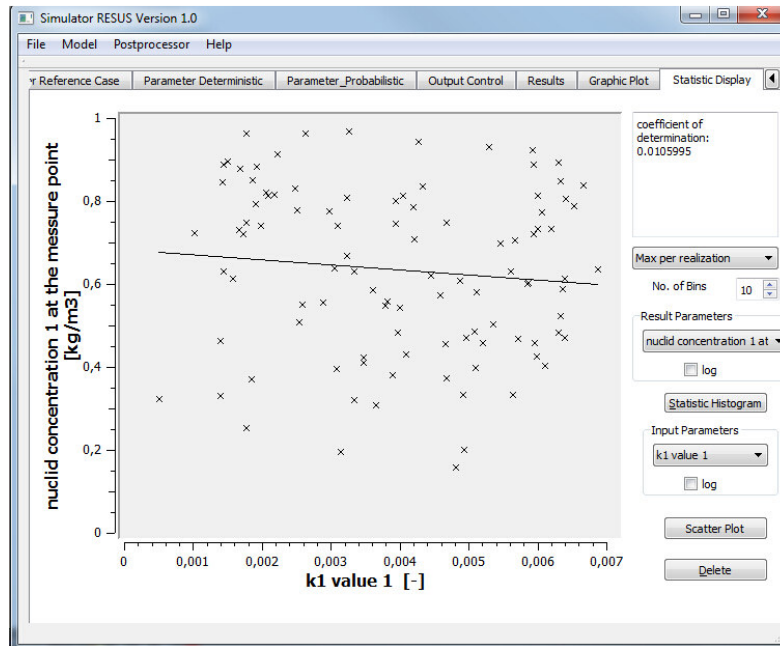


Abb. 4.68 Streudiagramm:  $K_d$ -Werte der Geoformation[-] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m<sup>3</sup>]

In Abb. 4.68 sieht man, dass die Punktwolke breit verteilt ist und das Bestimmtheitsmaß gering ist. Der Adsorptionseffekt ist bei diesem Transportbeispiel nicht mit der Nuklidkonzentration linear korreliert, und hat auf die Konzentration insgesamt nur geringe Auswirkungen. Dies war zu erwarten, da der Adsorptionskoeffizient aus didaktischen Gründen (s. oben) absichtlich niedrig angegeben wurde (gleichverteilt zwischen 0,001 bis 0,007 kg/kg) und wegen der geringen Modellgröße (100 m × 5 m) ohnehin einen beschränkenden Effekt auf die Konzentration hat.

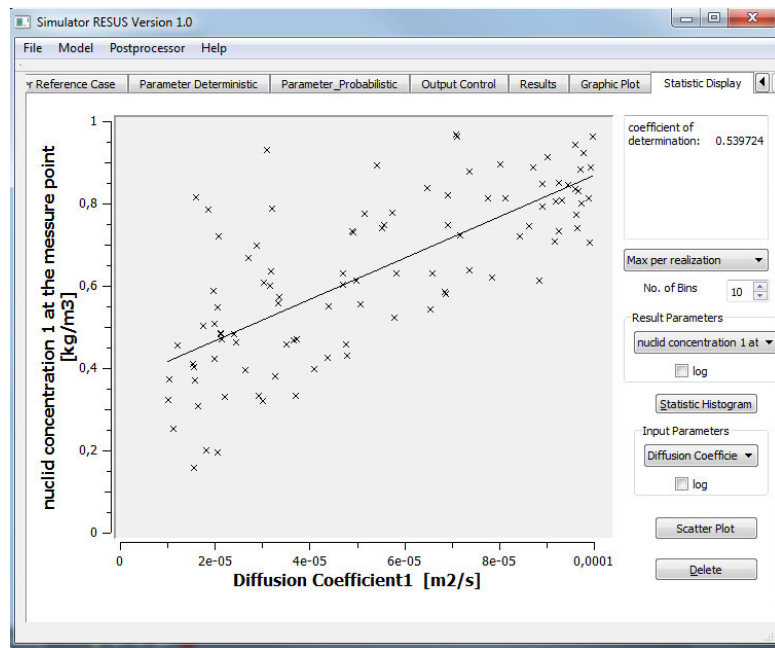


Abb. 4.69 Streudiagramm: Diffusionskoeffizient [ $\text{m}^2/\text{s}$ ] gegen die Nuklidkonzentration von U-235 am Messpunkt [ $\text{kg}/\text{m}^3$ ]

Zuletzt ist die Korrelation zwischen dem Diffusionskoeffizienten und der Konzentration des Radionuklids von Interesse. Anhand der Abb. 4.69 erkennt man eine relativ klare lineare Beziehung zwischen Diffusionskoeffizient und Konzentration am Messpunkt, die Punktwolke ist nah an der Regressionsgeraden mit dem Bestimmtheitsmaß = 0,540 angegeben. Da die Wasserströmung in die Y-Richtung sehr gering ist und der Stofftransport entlang der X-Richtung hauptsächlich durch Diffusionsprozesse verursacht wird, spielt der Diffusionskoeffizient hier neben der Permeabilität eine entscheidende Rolle im Transportprozess.



## **4.5 Beispiel für die Erstellung eines Analysemodells mit der dateibasierten Methode**

### **4.5.1 Allgemeine Vorstellung der dateibasierten Modellerstellungsmethode**

Neben der GUI-basierten Modellerstellungsmethode ist die sogenannte dateibasierte Methode das zweite mögliche Modellerstellungsverfahren in RESUS. Es wurde in erster Linie für einen Einsatz in der Forschung entwickelt und soll dem Benutzer eine deutlich größere Freiheit bei der Gestaltung und dem Aufbau der Rechenmodelle bieten. Die dateibasierte Methode ist in RESUS die Grundlage für die Erstellung von integrierten Modellen. Mit ihr können ganz beliebige Submodelle aus Fremdcodes geschaffen werden, im Gegensatz zu der GUI-basierten Methode ist sie nicht beschränkt auf die in RESUS fest implementierten Modelleinstellungen. Für die Benutzung der Methode sind jedoch deutlich weitgehendere Vorkenntnisse über die Arbeitsweise der angewendeten Fremdcodes notwendig. Diese können beim Einsatz von RESUS in Lehre und Ausbildung nicht vorausgesetzt werden, was den Anwendungsbereich dieser Methode auf die Forschung und auf den professionellen Bereich reduziert. Die dateibasierte Modellerstellungsmethode ist in dieser Version von RESUS nur einsetzbar, wenn die Eingabedateien der numerischen Fremdcodes textförmig (ASCII) sind. Bei binären Eingabedateien ist die Identifikation und Manipulation der Parameter problematisch.

In diesem Kapitel soll die dateibasierte Modellerstellungsmethode anhand eines Beispielmodells vorgestellt werden. Nach dieser Beschreibung des Modells wird erklärt, wie in RESUS die Konstruktion eines solchen Modells mit Submodellen erfolgt.

### **4.5.2 Beispielmmodell zur Erläuterung der dateibasierten Methode in RESUS**

In diesem Unterkapitel werden die Gegebenheiten des Beispielmodells beschrieben, das später mit Hilfe der dateibasierten Modellerstellungsmethode in RESUS nachgebildet werden soll. Ziel ist letztendlich, diese Methode zu veranschaulichen.

Als Beispielmmodell konstruieren wir einen Transportprozess für ein Endlager. Wir nehmen darin an, dass die Radionuklide (Schadstoffe) die technischen Barrieren des Endlagers durchdrungen haben und im umgebenden Wirtsgestein mit Grundwasser in Kontakt kommen. Die Radionuklide werden im Grundwasser gelöst und durch die Grundwasserströmung transportiert. Sie gelangen auf diese Weise nach und nach in verschiedene Bereiche: Ausgehend vom Wirtsgestein werden sie in das höherliegende Deckgebirge und schließlich in eine Trinkwasserquelle transportiert. Die letztgenannte Quelle liegt am Ende des Modells und aus ihr werden die Schadstoffe vom Menschen inkorporiert. Die Modellierung und Berechnung des gesamten Beispielprozesses erlaubt schließlich eine Abschätzung, wie wahrscheinlich und wie

stark unter den gegebenen Randbedingungen und Zufallswerten der Einfluss eines Radionuklidaustritts auf den Menschen ist und ob Grenzwerte eingehalten werden.

Die Bereiche, die der Schadstoff durchläuft, werden in unserem Modell in drei Teilen abgebildet. Jeder Teil hat andere Eigenschaften, die hier kurz umrissen werden sollen. Der erste beschreibt den Bereich des Wirtsgesteins. Für ihn wird hier Granit gewählt, das zu den Kristallingesteinen gehört [49].

Wir wollen die Modellteile im Hinblick auf die spätere Modellierung in RESUS als sogenannte „Submodelle“ bezeichnen. Die Modellierungslogik gebietet für die Berechnung verschiedener Teile auch verschiedene Rechenalgorithmen, die jeweils ein Submodell betreffen. Bevor aber eine Modellierung in RESUS möglich ist, müssen die Geometrie und Randbedingungen der Submodelle genauer festgelegt werden, was nachfolgend für die einzelnen Submodelle geschieht.

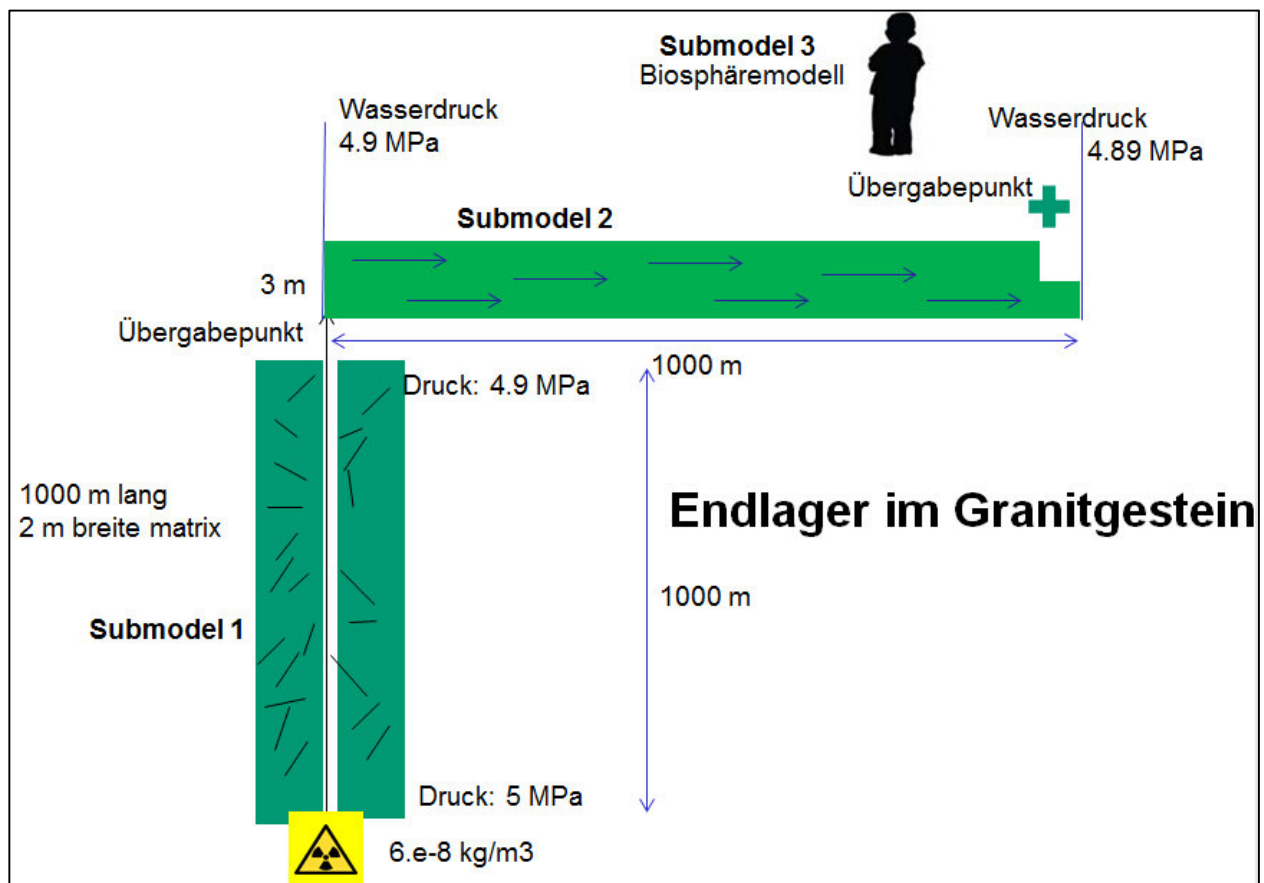


Abb. 4.70 Grafische Darstellung des Modellaufbaus mit drei Submodellen

Im Beispiel (siehe Abb. 4.70) ist das erste Submodell ein sogenanntes Kluft-Matrix-Modell im Granit-Wirtsgestein. Dieses besteht aus einem fast wasserdichten Material mit sehr niedriger Porosität und Permeabilität und ist mit Klüften durchzogen, die wasserdurchlässiger als die Gesteinsmatrix sind. Der Transport des Radionuklids findet hier ausschließlich in den Klüften durch die Grundwasserströmung statt. Darin durchdringt eine einzelne durchgehende

Hauptkluft von 1000 m Länge eine 2 m breite Granitmatrix, in der sich außerdem noch „dead-end“ Klüfte befinden. Die Hauptkluft wird in der obigen Abb. 4.70 als Linie dargestellt. Grundwasser strömt in ihr nach dem Darcy'schen Gesetz aufgrund eines hier vorgegebenen hydraulischen Gradienten von 0,1 MPa aus der Tiefe in das Deckgebirge (Submodell 2). Am Anfang der Hauptkluft tritt das Nuklid in den Grundwasserstrom ein und wird von diesem transportiert. Entlang des Transportweges kommt es zu Matrixdiffusion in die anliegenden Klüfte, die im Modell berücksichtigt ist. Die Granit-Matrix des Wirtsgesteins wird hier faktisch als wasserundurchlässig betrachtet, sie hat eine um Größenordnungen niedrigere Porosität und Permeabilität als der Kluftbereich. Betrachtet wird in diesem Beispiel in allen Submodellen die Ausbreitung eines einzigen Beispielnuklids Cl-36.

Das zweite Submodell (Submodell 2) stellt die Transportprozesse im aufliegenden Deckgebirge nach. Es besteht aus einem vereinfachten homogenen porösen Medium. Es ist ein zweidimensionales Transportmodell, das die Ausbreitung des Beispiel-Radionuklids in einem flächigen Bereich berechnet. Ursache der Ausbreitung ist auch hier im Wesentlichen die Grundwasserströmung. Diese ist durch einen Druckgradienten gegeben, die Flussrichtung ist im Modell von links nach rechts gerichtet. Wie in der Abb. 4.70 gezeigt, hat das Submodell 2 an seiner linken unteren Ecke einen Kopplungspunkt zum Submodell 1. Die dort als Punktquelle in das Submodell 2 eintretende Radionuklidmenge wurde im Submodell 1 als austretende Menge berechnet. Am Übergabepunkt, der rechts oben in der Abb. 4.70 des Submodells 2 gelegen ist, wird wiederum die hier austretende Nuklidkonzentration festgestellt. Dieser Wert geht schließlich als Eingangsgröße in das dritte Modell (Submodell 3) ein.

Im dritten unseres Modells berechnet nicht mehr den Transport, sondern lediglich den Dosiswert des Schadstoffs aus den Ergebnissen des jeweiligen Rechendurchlaufs. Dabei handelt es sich um ein Biosphärenmodell, das den Übergang des Grundwassers in eine Trinkwasserquelle nachstellt. Berechnet wird hier mit Hilfe der freigesetzten Konzentration des Nuklids angegeben im für den Menschen relevanten Dosiswert (Siehe Gleichung 4-13), der auch in Level-E verwendet wird [24].

$$Dose(t) = \frac{p}{w} \beta_i C_i(t) \quad (4-13)$$

$w$  ist hier die Wassermenge [ $m^3/Jahr$ ] des betrachteten Flusses;  $\beta_i$  ist der jeweilige Dosisfaktor der Radionukliden,  $p$  [ $m^3/Jahr$ ] ist die von einem Menschen im Jahr getrunkene Wassermenge.  $C_i(t)$  [ $kg/kg$ ] ist die zeitabhängige Konzentration des betrachteten Nuklids im Flusswasser.

Das Beispielmmodell ist damit hinlänglich umrissen. Bevor mit ihm in RESUS ein Modell erstellt wird, wenden wir uns der zunächst der dafür notwendigen Modellierungslogik zu. Sie ist der Kern dieses Kapitels, ihr grundlegendes Verständnis ist wesentlich für die spätere Benutzung der dateibasierten Methode. Die hier erläuterten Abläufe sollen auf vereinfachte Weise vermitteln, auf welche Weise RESUS integrierte Modelle aus mehreren Submodellen verarbeitet. Deren Berechnung findet üblicherweise in Submodellen statt.

Das bedeutet hier, jedes der drei Submodelle wird separat in jeweils einer Komponente modelliert, um anschließend in einer Modellkette verknüpft zu werden (siehe Abb. 4.70).

Zum Verständnis der Arbeitsabläufe der dateibasierten Methode ist grundlegendes Wissen über die Kommunikation von RESUS mit den numerischen Fremdcodes von Vorteil. Die Vorgehensweise des Programms sollen hier kurz umrissen werden, sie wird in Kapitel 5 genauer ausgeführt. Ein numerischer Fremdcode benötigt, ähnlich wie RESUS bei der Modellerstellung, mindestens eine Datei, welche die wesentlichen Daten für die durchzuführende Berechnung enthält. Diese nennen wir „Eingabedatei“. Über diese Eingabedatei läuft die gesamte Kommunikation von RESUS mit dem Fremdcode. Die Eingabedatei wird von RESUS bei jedem Rechendurchlauf automatisch erstellt und an den Fremdcode weitergegeben. Diesem werden darin unter anderem die Parameterwerte für die Rechnung und die auszugebenden Parameter mitgeteilt. RESUS ändert für jeden Rechendurchlauf des Fremdcodes die probabilistischen Parameterwerte in der Eingabedatei. Sie werden entsprechend ihrer festgelegten Verteilung zufallsgeneriert und an der vorgesehenen Stelle in die Eingabedatei geschrieben. Für den ersten Rechendurchlauf einer Modellrechnung werden, wie schon oben erwähnt, für alle probabilistischen Parameter Referenzwerte eingesetzt, die vom Benutzer am Anfang der Modellerstellung angegeben werden. Diese sind ebenfalls in den Eingabedateien enthalten.

Die Template-Eingabedatei ist eine vom Benutzer zu erstellende Vorlage für die Eingabedatei, die noch zusätzliche Information zum Modellaufbau für RESUS enthält. Wie oben erwähnt ist die dateibasierte Modellerstellungsmethode in dieser Version von RESUS nur einsetzbar, wenn die Eingabedateien der numerischen Fremdcodes textförmig (ASCII) sind. Bei binären Eingabedateien ist die Identifikation der probabilistischen und Eingabeparameter problematisch.

Neben den Parameterwerten sind in der Template-Eingabedatei auch entscheidende Angaben zum Modellaufbau hinterlegt. Diese betreffen die Verknüpfungen der Parameter von einem Submodell mit dem nächsten. Die Template-Eingabedatei eines Fremdcodes gibt RESUS die Information, welche ihrer Ausgabeparameter an den nächsten Fremdcode weiterzugeben sind, um dort in die Berechnung einzugehen.

Um ein Analysemodell mit Hilfe der dateibasierten Methode herzustellen, muss zuerst eine Beschreibungsdatei verfasst werden. Die Beschreibungsdatei ist eine Text-Datei, die von Nutzer in ASCII-Text gespeichert wird. Sie enthält Informationen zu den wichtigsten Modelleigenschaften des Hauptmodells und der Submodellen, aus denen RESUS die Modellstruktur erstellt. Die Beschreibungsdatei hat eine festgelegte Struktur, damit RESUS die Informationen korrekt einliest. Die Reihenfolge der Information in der Datei ist in Blöcke unterteilt. Im ersten Block stehen Modellname, Modelltyp, Modellbeschreibung und die Anzahl der Submodelle. Die danach folgenden Modellblöcke sind immer gleich aufgebaut und betreffen jeweils ein Submodell. Sie enthalten die Angaben Submodellname, Submodelltyp, Submodellbeschreibung und die Namen der Template-Eingabedateien des entsprechenden Submodells (siehe Abb. 4.71). Die Reihenfolge der Submodellblöcke bestimmt die Berechnungsreihenfolge der Submodelle und damit die Struktur des Gesamtmodells.

Der Erstellungsvorgang läuft im Groben wie folgt ab: Der Benutzer erstellt eine für RESUS spezifische Beschreibungsdatei, welche Grundinformation über das zu erstellende Modell beinhaltet, z.B. Modellname, Modelltyp, Submodellanzahl usw. Zudem muss für jedes Submodell eine eigene Template-Eingabedatei erstellt werden, in der Informationen für die

Submodelle hinterlegt wird. Mit der Grundinformation der Beschreibungsdatei öffnet und liest RESUS die Template-Eingabedateien jedes Submodells. Die eingelesene Information wird an die RESUS-GUI weitergeleitet, wo sie weiter bearbeitet (Werteinstellung, Samplegenerierung, Einstellung der Ausgabeparameter usw.) und zur Simulation und Analyse verwendet werden kann.

Die Beschreibungsdatei in unserem Beispiel wird „RESUS\_Example“ genannt (Abb. 4.71).

```
The model name is:
RESUS Example
The model type is:
using Rockflow entire input data with submodels
The model description:
This is an Example of RESUS for a pseudus granite diposal
The number of submodels of this model calculation:
3

The sub-model name is:
fracture-Matrix transport
The model type is:
using RockFlow solver
The model description:
host rock transport model for granite disposal
The name of the input file of the submodel 1:
tangV1.rfd

The sub-model name is:
geosphere Transport 2D
The model type is:
using RockFlow solver
The model description:
Overburdens transport model
The name of the input file of the submodel 2:
asm2d.rfd

The sub-model name is:
Biosphere model
The model type is:
LevelE Biosphere
The model description:
Biosphere model of Level-E
The name of the input file of the submodel 3:
Biosphere.txt
```

Abb. 4.71 Beschreibungdatei unseres Modellbeispiels, benannt „RESUS\_Example“

Im Folgenden soll die genaue Vorgehensweise der dateibasierten Modellerstellung in RESUS erläutert werden. Diese beginnt wieder im „Model Selection“-Tab für die Modellauswahl. Aus der Baumstruktur wird der Punkt „File based construction“ und dessen Unterpunkt „Submodel-Chain“ mit Doppelklick ausgewählt. Daraufhin erscheint ein Fenster, in dem der Ort der Beschreibungsdatei ausgewählt werden muss (siehe Abb. 4.72). RESUS liest diese

Beschreibungsdatei ein und baut automatisch die darin definierte Modellstruktur auf. Weiterhin werden die mit darin bezeichneten Template-Eingabedateien der Submodelle eingelesen.

Die probabilistischen Parameter werden in den Template-Eingabedateien mit „SIMULATE“ gekennzeichnet. Der Referenzwert des Parameters steht hinter dem „SIMULATE“. Es können an dieser Stelle zusätzlich Information zu den Parametern hinterlegt werden, genauer: der Parametername, die Parametereinheit, die statistische Verteilung der Parameter und die statistischen Verteilungsmaße. Das Element „RESULT“ kennzeichnet die Ausgabeparameter, dahinter schreibt man die Namen der Dateien, in die die Ergebnisse geschrieben werden sollen; den Name des Ausgabeparameters; Einheit des Ausgabeparameters; die Beschreibung des Parameters; der Check „ifCouple“. „ifCouple“ entscheidet ob der Ausgabeparameter als Eingabe für das nächste Submodell genutzt werden soll, dabei steht 1 für „true“ und 0 steht für „false“.

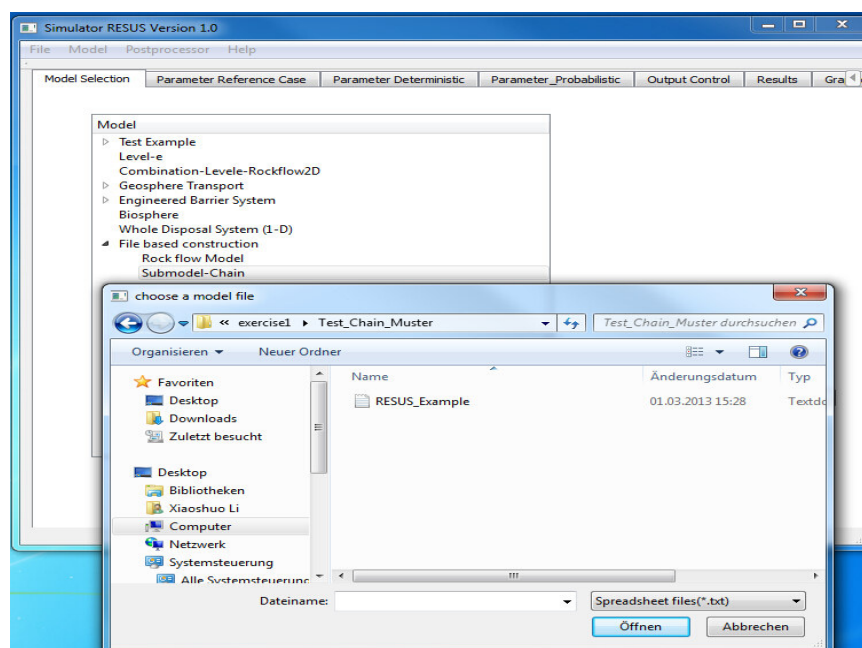


Abb. 4.72 Modellauswahl beim ersten Tab für das dateibasierte Modellerstellungsverfahren

```
#OUTPUT_EX

;RESULT
tangv1.plt

CONC  ; kg/m3  Concentration nuclide1 in model 1  1

#SOIL_PROPERTIES

; SIMULATE

0.35  ; porosity_matrix - uniform  0.2  0.5
```

Abb. 4.73 Beispiel der Template-Eingabedatei, in der ein probabilistischer Parameter und ein Ausgabeparameter gekennzeichnet sind

Abb. 4.73 zeigt einen Abschnitt in der Template-Eingabedatei, in der die probabilistischen Parameter und die Ausgabeparameter gekennzeichnet werden. RESUS liest die Template-Eingabedateien ein, erkennt die gekennzeichneten Parameter und fügt die entsprechende Information dem Modell hinzu. Nach der Erstellung des Modells springt die RESUS-GUI in den zweiten Tab, wo sich die Einstellungstabelle für die Referenzwerte der probabilistischen Parameter befindet. Die probabilistischen Parameter und die in der Template-Eingabedatei angegebenen Werte werden in der Tabelle bereits automatisch eingetragen, der Nutzer kann die Werte direkt übernehmen lassen oder eine Korrektur für die Referenzwerte in der Tabelle durchführen (Siehe Abb. 4.74). Es gibt 11 probabilistische Parameter in diesem Beispiel. Die in unserem Modell hinterlegten Werte der Referenzwerte für die Parameter sieht man in der Tabelle 4-9.

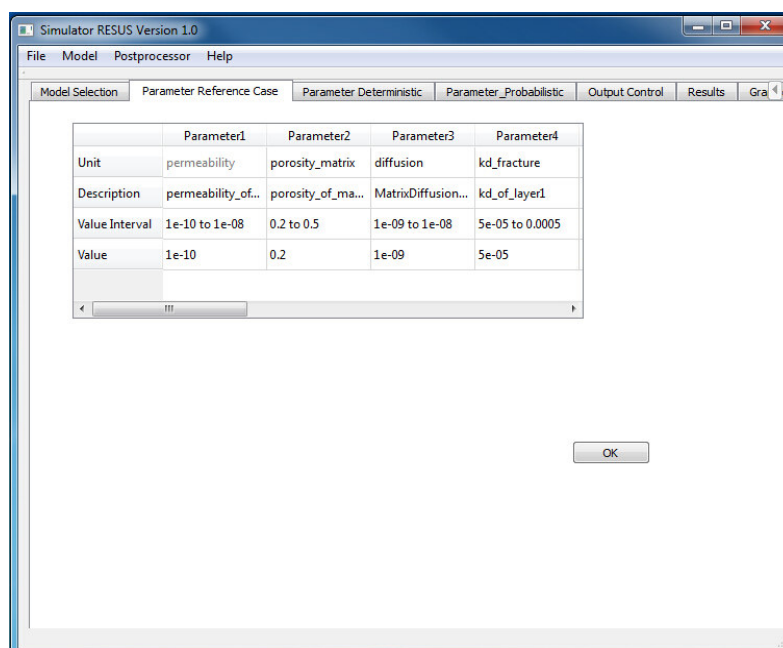


Abb. 4.74 Tabelle im Tab für die Einstellung der Referenzwerte für die probabilistischen Parameter

Tabelle 4-9 Einstellung der Referenzwerte der probabilistischen Parameter

Parametername	Einheit	Wert
Permeabilität der Kluft in Submodell1	$m^2$	$8 \cdot 10^{-10}$
Porosität der der Matrixphase in Submodell1	-	0,35
Diffusionskoeffizient des Schadstoffs in Submodell1	$m^2/s$	$1 \cdot 10^{-8}$
Kd-Wert der Kluft in Submodell1	$kg/kg$	$5 \cdot 10^{-5}$
Porosität der Kluft in Submodell1	-	0,8

Parametername	Einheit	Wert
Dispersionslänge in Submodell1	$m$	5,0
Porosität des Deckgebirges in Submodell2	-	0,2
Kd-Wert des Deckgebirges in Submodell2	$kg/kg$	$5 \cdot 10^{-5}$
Permeabilität des Deckgebirge in Submodell2	$m^2$	$1 \cdot 10^{-12}$
Dispersionslänge des Deckgebirges in Submodell2	$m$	5,0
Volumenstrom des Flusses in Biosphären Modell	$m^3/a$	$1 \cdot 10^7$

Entsprechend der Tabelle 4-9 werden die Referenzwerte für die Berechnung verwendet. Da die Werte durch die Angaben in den Template-Eingabedateien festgelegt werden, wird der Tab für deterministische Parameter „Parameter Deterministic“ nicht benötigt, also kann die Benutzeroberfläche direkt zu „Parameter\_Probabilistic“ übergehen. In der Tabelle dieses Tabs sind die probabilistischen Parameter wie bei den anderen Modellen aufgelistet. Der Benutzer kann entweder die Verteilungseinstellung aus der Template-Eingabedatei übernehmen lassen, oder an dieser Stelle Korrekturen der übernommenen Einstellungen anbringen (siehe Abb. 4.75). Die Anzahl der Zufallswerte soll 300 betragen. Die statistischen Verteilungen der Parameter werden nach Tabelle 4-10 eingestellt.

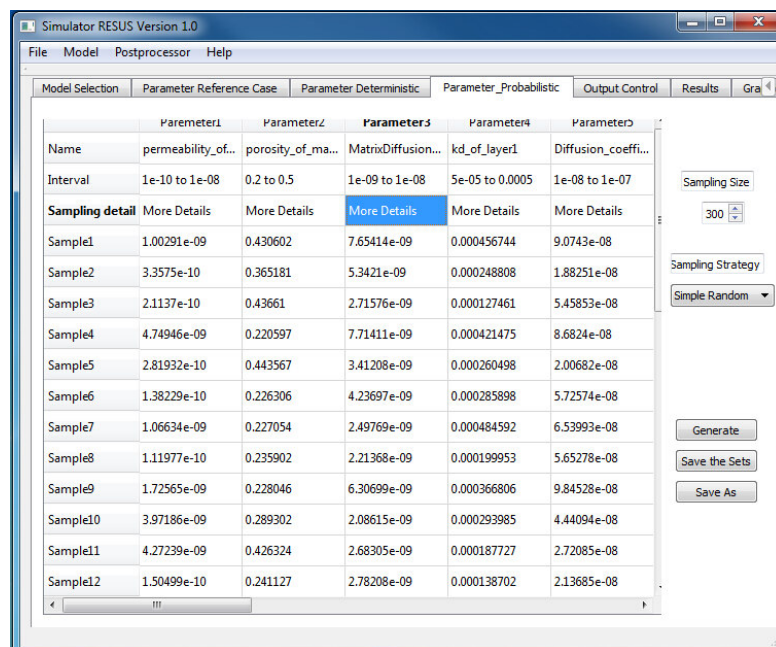


Abb. 4.75 „Parameter\_Probabilistic“ Tab auf der RESUS-GUI für die Korrekturen der statistischen Verteilungen der Parameter



Tabelle 4-10 Verteilungseinstellung für die probabilistischen Parameter

Parametername	Statistische Verteilung	Verteilungsmaße
Permeabilität der Kluft in Submodell 1	Log-Gleichverteilung	$1 \cdot 10^{-10} - 1 \cdot 10^{-8}$
Porosität von der Matrixphase in Submodell1	Gleichverteilung	0,2 – 0,5
Diffusionskoeffizient des Schadstoffs im Submodell1	Gleichverteilung	$1 \cdot 10^{-5} - 5 \cdot 10^{-5}$
Kd-Wert der Kluft in Submodell 1	Gleichverteilung	$5 \cdot 10^{-5} - 5 \cdot 10^{-4}$
Porosität der Kluft in Submodell1	Gleichverteilung	0,8 – 1,0
Dispersionslänge in Submodell1	Gleichverteilung	5,0 – 10,0
Porosität des Deckgebirges in Submodell2	Gleichverteilung	0,05 – 0,4
Kd-Wert des Deckgebirges in Submodell2	Gleichverteilung	$5 \cdot 10^{-05} - 5 \cdot 10^{-04}$
Permeabilität des Deckgebirges in Submodell2	Log-Gleichverteilung	$1 \cdot 10^{-13} - 1 \cdot 10^{-11}$
Dispersionslänge des Deckgebirges in Submodell2	Gleichverteilung	5,0 – 10,0
Volumenstrom des Flusses in Biosphären Modell	Deterministisch	$1 \cdot 10^{+7}$ als Wert

Nach der Einstellung der Verteilungen kann der Nutzer mit Druck auf den Knopf „Generate“ die 300 Zufallswerte entsprechend der Verteilungen generieren. Nach der Erzeugung dieser Samples ist das Modell bereit für die Berechnung.

Der nächste Schritt betrifft den letzten Tab zur Einstellung des Pre-Processings, namens „Output Control“. Ähnlich wie bei den bereits vorgestellten Modellen wird hier festgelegt, welche der errechneten Modellparameter ausgegeben werden sollen. Bei unserem Beispiel gibt es drei Ausgabeparameter, die in einer Tabelle aufgeführt sind: Die Nuklidkonzentration am Kluftende in Submodell 1, die Nuklidkonzentration des Messpunkts in Submodell 2 und der Schadstoff-Dosiswert im Biosphärenmodell (Submodell 3). Alle Parameter sollen in diesem Beispiel dargestellt werden, in der „Output“-Spalte wird also „y“ eingestellt (siehe Abb. 4.76).

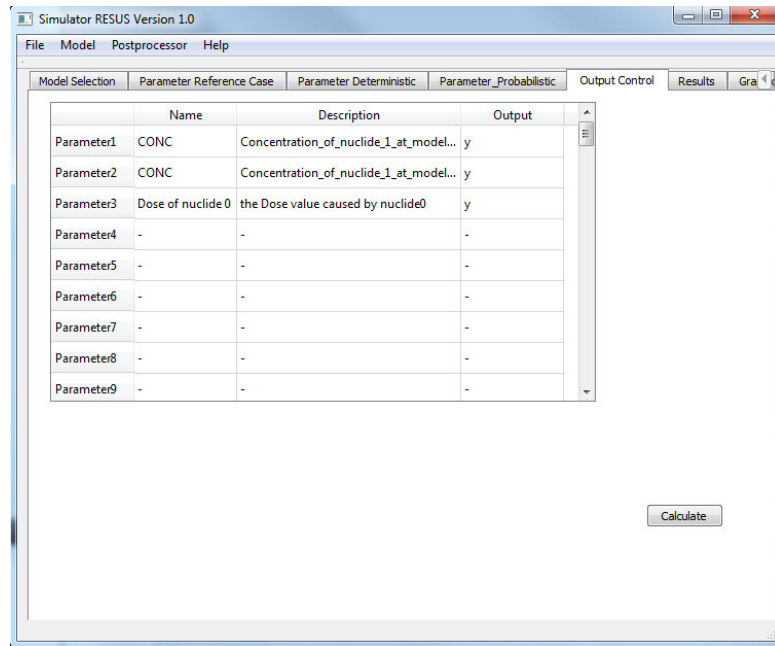


Abb. 4.76 Tab „Output\_Control“ zur Festlegung, welche Ausgabeparameter in diesem Beispiel weiter verarbeitet werden

Um die 300 Durchläufe der Modellberechnung durchführen zu lassen klicken wir auf „Calculate“. Die Berechnung dauert auf einem PC mit 3,00GHz Intel Core 2 Duo Prozessor ungefähr 3 Stunden.

Nach der Kalkulation kann das Post-Processing, also die Auswertung der durchgeführten Modellrechnungen, beginnen. Die Vorgehensweise in diesem Schritt ist unabhängig von der Modellerstellungsmethode (dateibasiert oder GUI-basiert) und wurde in den vorherigen Kapiteln bereits ausführlich erläutert. Wir wenden uns im folgenden Auswertungsbeispiel daher weniger der Vorgehensweise, sondern vor allem der logischen Verknüpfung der verschiedenen Auswertungsmethoden auf RESUS zu. Beispielhaft liefern wir mögliche Interpretationsansätze für die interessanteren Ergebnisse aus der Berechnung unseres Beispiels.

Um eine Übersicht zu gewinnen, betrachten wir zunächst die zeitlichen Verläufe der Schadstoffdosis im Biosphärenmodell. Dies erfolgt im Tab „Graphic plot“ der RESUS-GUI. Abb. 4.77 zeigt für unsere Rechnung das Diagramm der 300 Dosiswert-Verläufe über den betrachteten Zeitraum von 300000 Jahren. Zudem werden die statistischen Kennwerte (Minimalwert, 25%-Quantil, Median, 75%-Quantil, Maximalwert, Mittelwert) der Verläufe dargestellt. Wie zu sehen ist, liegen die Kennwerte in diesem Beispiel alle außer dem Maximum und Mittelwert übereinander auf der Null-Linie, da für mehr als 75% aller Realisierungen mit Null-Freisetzung errechnet wurden. Es erfolgte also bei dem Großteil der Rechenrealisierungen keine Nuklidfreisetzung in die Biosphäre. Der höchste errechnete Wert von allen Läufen tritt nach einem Zeitraum von 60000 Jahren auf und erreicht etwa  $2,3 \cdot 10^{-07}$  Sv/y.

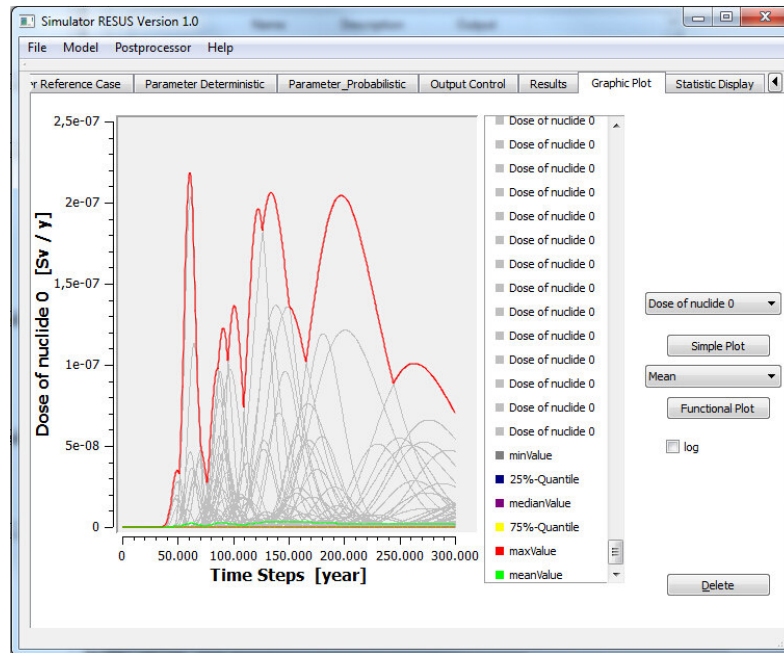


Abb. 4.77 Dosiswerte des Beispiels und deren statistischen Kennwerte aus 300 Realisierungen

Nach der Übersichtsdarstellung der zeitabhängigen Ergebnisse wird die genauere Analyse im Tab „Statistic Display“ durchgeführt. Zunächst wollen wir die Verteilung der maximalen Nuklidkonzentrationen der einzelnen Rechendurchläufe in einem Histogramm auswerten.

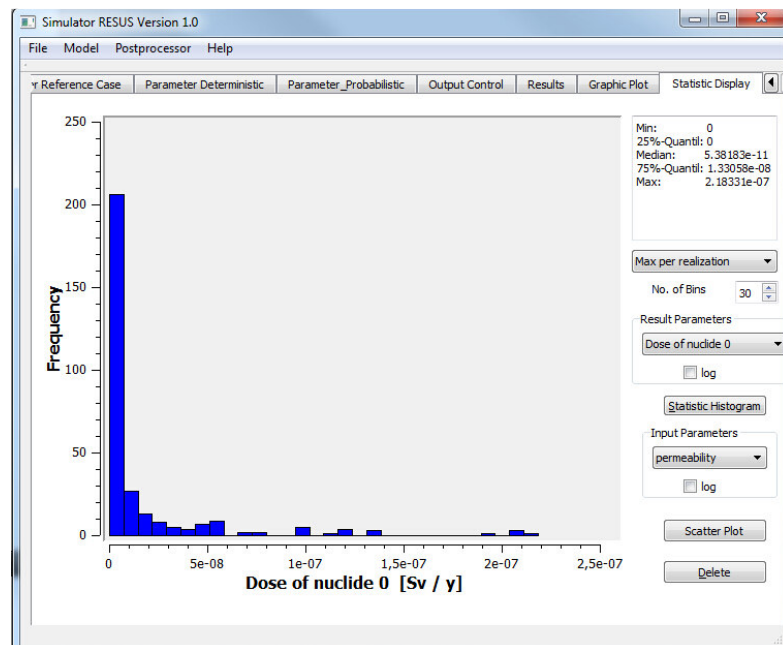


Abb. 4.78 Histogramm der maximalen Werte jeder Realisierung mit 30 Wertklassen

Abb. 4.78 zeigt das Histogramm für die maximalen Dosiswerte jeder Rechenrealisierung. Die höchste Säule links beschreibt die Realisierungen ohne Freisetzung, die, wie oben festgestellt, den Großteil ausmachen.

Im nächsten Schritt wollen wir eine Sensitivitätsanalyse mittels des Streudiagramms zwischen den Ein- und Ausgabedaten durchführen.

Abb. 4.79 zeigt den Zusammenhang zwischen den Dosis-Maximalwerten aller Realisierungen und den logarithmierten Permeabilitäten der Kluft im Submodell 1. Man kann eindeutig sehen, dass die Ergebnisse von der Permeabilität der Kluft stark beeinflusst (mit Bestimmtheitsmaß = 0,858) sind. Es beweist, dass der advective Transport im Submodell 1 das gesamte Modell dominiert.

Es ist aber auch interessant, dass der Scatterplot nach der Größe der Permeabilität in drei Bereiche geteilt ist. Bei niedrigerer Permeabilität ( $1 \cdot 10^{-10}$  bis  $1 \cdot 10^{-9,5} \text{ m}^2$ ) hat das Modell meistens eine Null-Freisetzung, da die Wasserströmung auch relativ gering ist. Bei höheren Permeabilitäten ( $1 \cdot 10^{-9}$  bis  $1 \cdot 10^{-8} \text{ m}^2$ ) ist der Einfluss der Permeabilität auf das Ergebnis relativ groß, wobei die Punktwolke sich in einer Linie konzentriert. Bei der Konzentration im mittleren Bereich ( $1 \cdot 10^{-9,5}$  bis  $1 \cdot 10^{-9} \text{ m}^2$ ) ist der Einfluss der Permeabilität auf das Ergebnis zwar immer noch stark sichtbar, aber der Einfluss der anderen Eingabeparameter ist in diesem Bereich auch stärker, sodass die Punktwolke in diesem Bereich breiter verteilt ist.

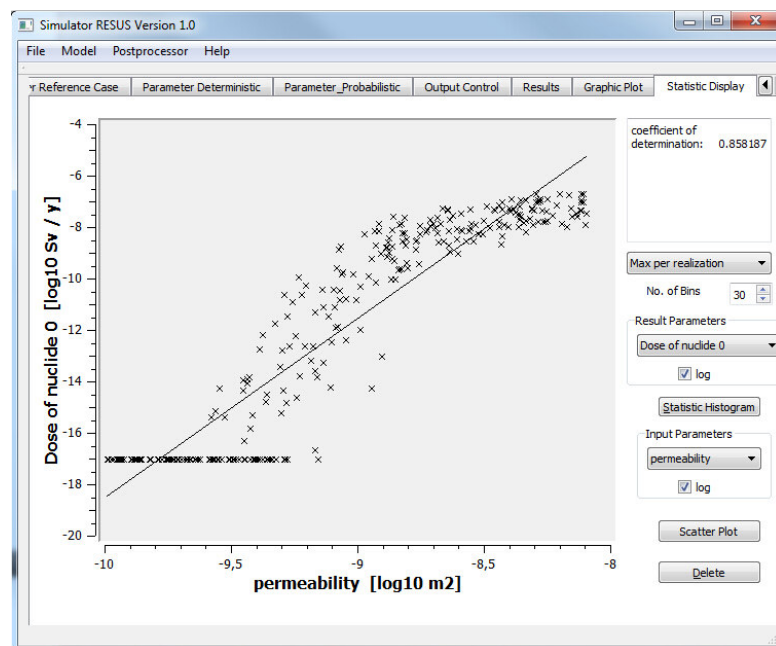


Abb. 4.79 Logarithmisches Streudiagramm: Dosiswerte gegen die Permeabilität der Kluft in Submodell 1

Die Null-Ausgaben werden auf den Wert ( $1 \cdot 10^{-17}$ ) gesetzt, da die Werte logarithmisch transformiert werden müssen. Da null- und negative Werte nicht logarithmisch transformiert

werden können, werden diese beim Streudiagramm durch den kleinsten transformierten Wert der Datenreihe ersetzt.

Nach dem advektiven Transport in der Kluft im Submodell 1 transportiert das Submodell 2 als Deckgebirgsmodell das Radionuklid weiter. Der Einfluss aus der Permeabilität des Deckgebirges im Submodell 2 ist in Abb. 4.80 zu sehen.

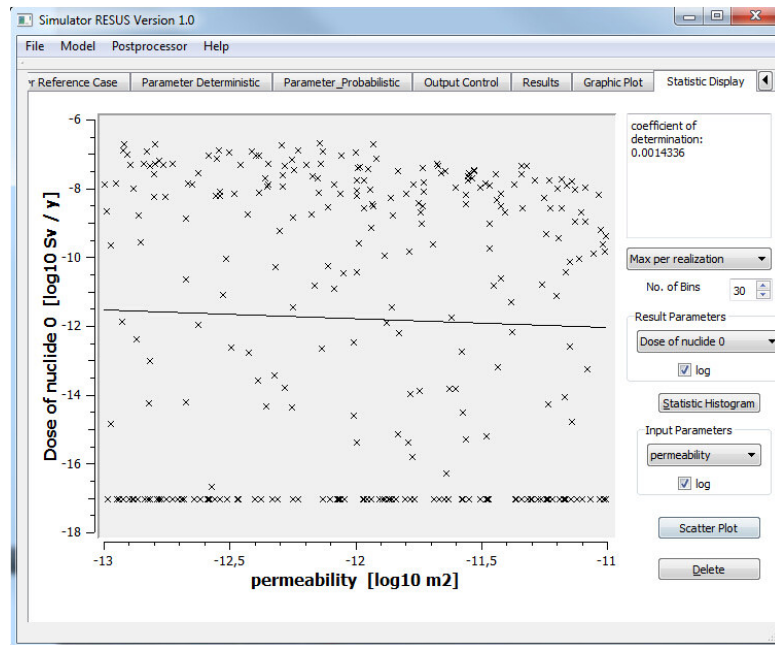


Abb. 4.80      Logarithmisches Streudiagramm: Dosis-Werte gegen die Permeabilität in Submodell 2

In Abb. 4.80 sieht man, dass die Permeabilität vom Deckgebirge keinen eindeutigen Einfluss auf Dosiswerte (Bestimmtheitsmaß = 0,001) hat. Dies liegt möglicherweise daran, dass die Ergebnisse von der Permeabilität im Submodell 1 so stark beeinflusst werden, dass die Einflüsse aus anderen probabilistischen Parametern geschwächt werden.

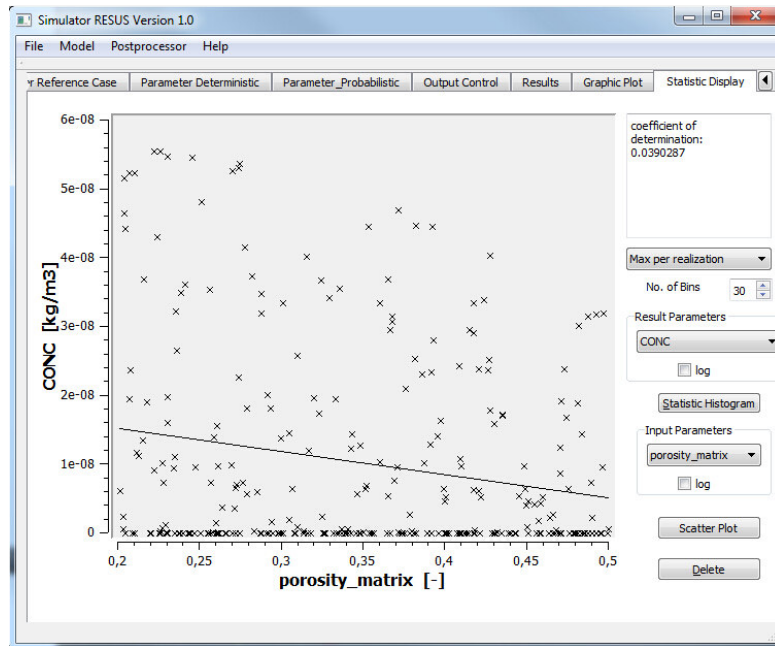


Abb. 4.81 Streudiagramm: maximale Konzentrationen am Kluftende gegen die Porosität der Matrix in Submodell 1

Es ist auch interessant zu sehen, wie groß der Effekt der Materialeigenschaft der Gesteinmatrix auf den Transport eines Radionuklids ist. Abb. 4.81 zeigt den Streudiagramm zwischen der Radionuklidfreisetzung am Kluftende und der Porosität der Matrix. Wegen der starken Dominanz des advektiven Transports in Submodell 1 ist es schwierig, den Effekt der Porosität der Matrix zu erkennen.

## 4.6 Speichern und Wiederöffnen eines Analysemodells in RESUS

Als eine Plattform zur Modellierung muss RESUS auch die Funktionalität anbieten, dass die mit RESUS erstellten und durchgerechneten Modelle in Dateien gespeichert und aus Dateien wiederhergestellt werden können.

### 4.6.1 Speichern eines Modells

Es gibt zwei Möglichkeiten, wie der Nutzer das Modell in RESUS-Dateien abspeichern kann. Die erste findet sich im Hauptmenü und die zweite auf dem „Parameter\_Probabilistic“ Tab. RESUS-Dateien sind die Dateien, die das in RESUS laufende oder gelaufene Modell speichern. Eine genaue Beschreibung des Formats der RESUS-Dateien findet sich in Kapitel 5.3.8.

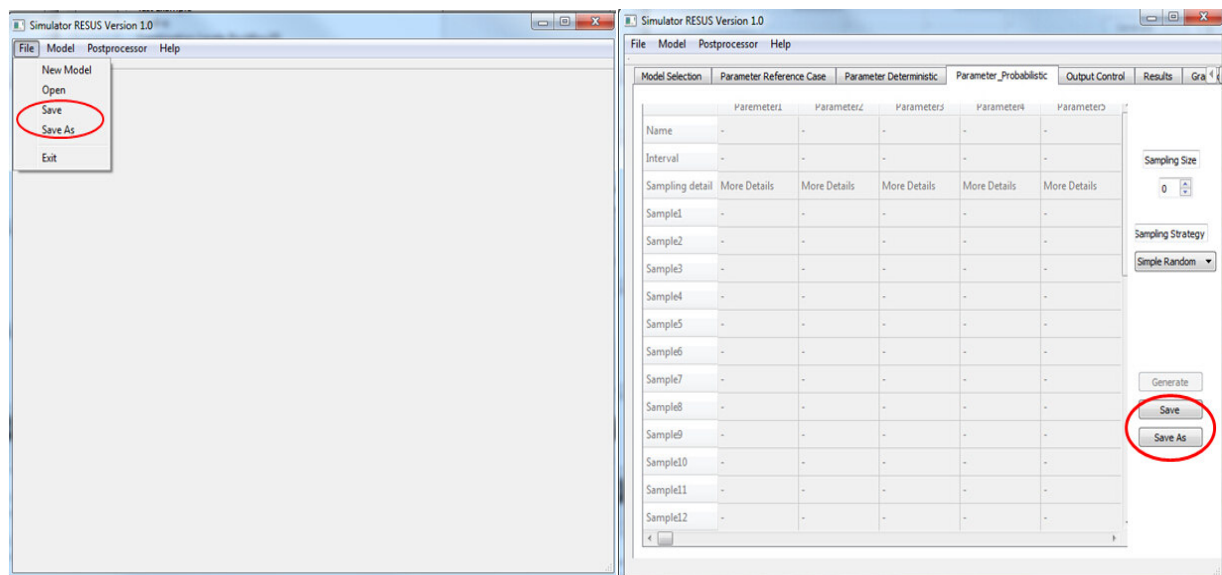


Abb. 4.82 Kommandostelle für das Speichern des auf RESUS laufenden Modells. Der rote Kreis markiert den Kommando-Knopf für das Speichern

Für das Speichern bietet RESUS immer zwei Optionen an, „Save“ und „Save As“. „Save As“ dient dazu, dass eine neue RESUS-Datei erzeugt wird, in die das aktuell laufende Modell gespeichert wird.

Wenn „Save As“ Knopf angeklickt wird, wird ein Fenster aufgerufen, um die Datei für das Speichern und deren Pfad festzulegen (siehe Abb. 4.83). Der Dateiname muss vom Nutzer eingegeben werden. Das Modell wird in die erzeugte Datei gespeichert. Wenn die Rechenprozesse des Modells durchgelaufen sind, werden die Ergebnisse der Modellrechnung in die Ergebnisdateien gespeichert.

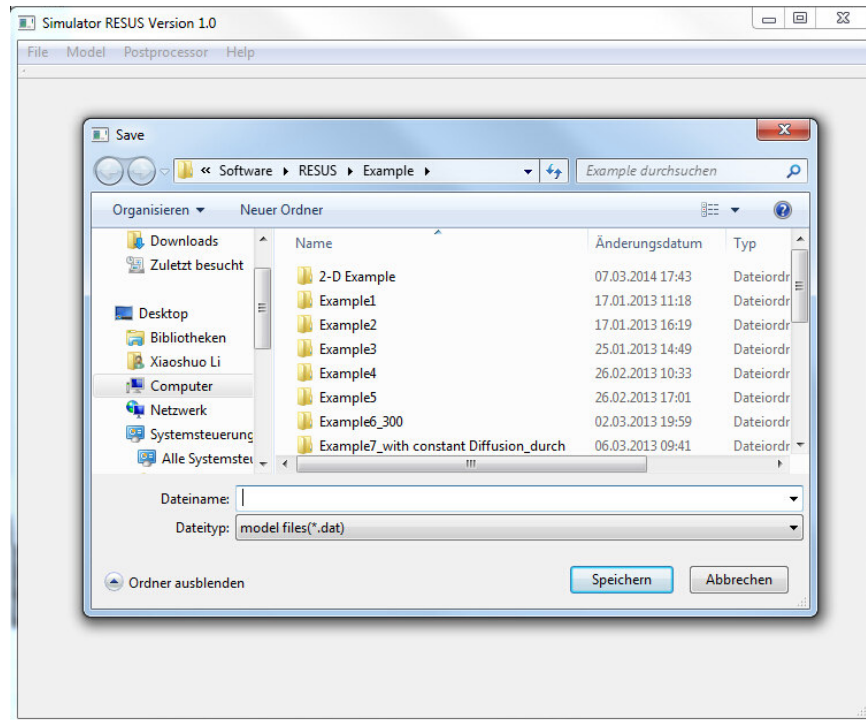


Abb. 4.83 Dialogfenster für die Dateiauswahl für die Speicherung des Modells

Die Option „Save“ dient dazu, dass die Modellinformation automatisch in die Datei gespeichert wird, aus der das Modell eingelesen wurde. Wenn das Modell neu erzeugt und nie gespeichert wurde, funktioniert „Save“ identisch wie „Save As“.

Wenn der Nutzer noch kein Modell erstellt hat aber trotzdem versucht zu speichern, ruft RESUS ein Warnungsfenster auf.

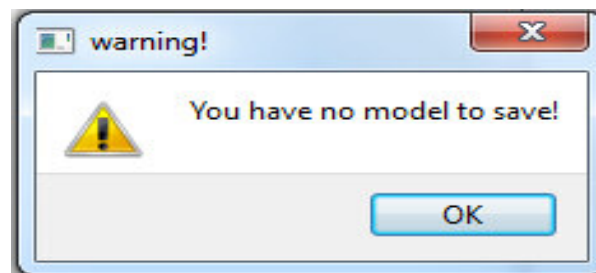


Abb. 4.84 Warnungsfenster beim Speicherversuch ohne erstelltes Modell



#### 4.6.2 Dateien zum Speichern der Modellinformationen

RESUS benötigt vier Dateien zum Speichern der Modellinformation. Eine ist die allgemeine Informationsdatei mit der Suffix „.dat“, in die die Attribute vom Modell und Parametern gespeichert werden. Der Name der Datei wird von Benutzer definiert (siehe Kapitel 4.6.1). Außerdem werden drei weitere Textdateien mit der Suffix „.txt“ automatisch erzeugt, die der Speicherung der Modellinformation dienen. Der vom Nutzer definierte Name wird mit bestimmten Worten automatisch ergänzt und es werden folgende neuen Dateien im gleichen Ordner erstellt:

Name\_Samplings.txt      gesampelte Daten für probabilistische Parameter als Matrix

Name\_Results.txt      allgemeine Informationen und Daten der Ergebnisparameter

Name\_SamResults.txt      errechneten Daten der Ergebnisparameter als Matrix

Das Format der Speicherdateien wird in Kapitel 5.3.8 dokumentiert. Alle diese Dateien sind im ASCII-Format erzeugt und deswegen vom Nutzer manipulierbar.

#### 4.6.3 Wiederöffnung eines Modells

Die gespeicherten Modelle können in RESUS wieder geöffnet werden. Es gibt zwei Möglichkeiten, wie Modelle von Dateien eingelesen werden können (siehe Abb. 4.85).

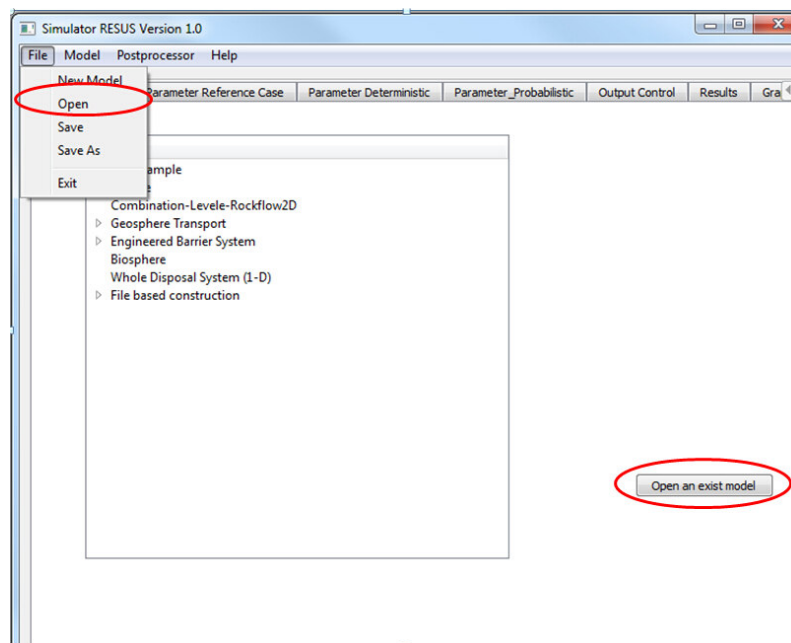


Abb. 4.85 Menüeintrag für die Wiederöffnung eines Modells

Beim Anklicken von „open“ wird ein Dialogfenster aufgerufen, wobei der Nutzer das bereits erstellte Modell finden und öffnen kann. Im Fenster sind nur die „.dat“-Dateien sichtbar. Der Nutzer braucht nur die „.dat“-Datei zu wählen, RESUS sucht automatisch die entsprechenden weiteren „.txt“-Dateien, um die Daten und Information des Modells wieder einzulesen. Nach dem Einlesen werden alle Informationen in den Tabellen dargestellt. Falls das Modell durchgerechnet worden ist, ist RESUS dann auch bereit, die Ergebnisse mit allen seinen Darstellungsmethoden zu analysieren.

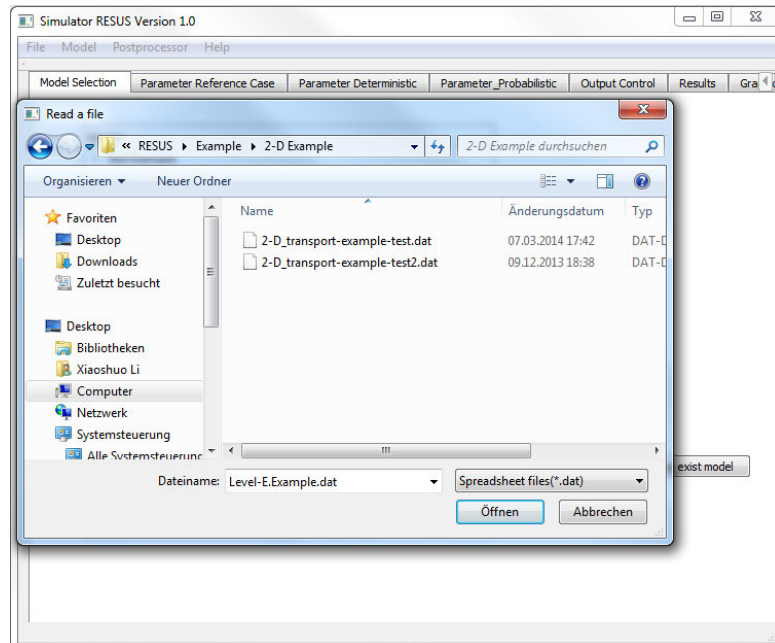


Abb. 4.86 Dialogfenster Zum Öffnen, Wiederherstellen, Einlesen eines Modells

## 5 Entwicklungsdokumentation des Programms RESUS

RESUS wird in der objekt-orientierten Programmiersprache C++ [49] entwickelt. Sämtliche Behandlung der Daten, wie zum Beispiel der Modell-, Parameter- und Analyseinformation erfolgt daher in objektorientierter Form. Die Objekte für das Modell können während oder nach der Modellanalyse in Dateien gespeichert und nachträglich wieder in RESUS aufgerufen werden. Die Objektstruktur und Modellspeicher sowie deren Wiedererstellung werden in den folgenden Unterkapiteln beschrieben.

Die in diesem Kapitel beschriebene Funktionsweise und der theoretische Hintergrund der RESUS-Software betrifft allgemein die Erstellung und Berechnung der Modelle. Dabei sind die grundlegenden Funktionsweisen des Programms für die GUI-basierte Methode und die dateibasierte Modellerstellungsmethode gleich.

### 5.1 Details der Klassen- und Objektsstruktur

Mithilfe des Qt-Designers [17] und Qt-Creators [51] lässt sich eine Bedienungsoberfläche (GUI) interaktiv erstellen und die einzelnen Schaltelemente mit C++ Funktionsaufrufen verknüpfen, so dass man ein ausführbares Programm erhält.

Die Modellinformation wird mit einer Klassen- und Objektsstruktur definiert und gespeichert.

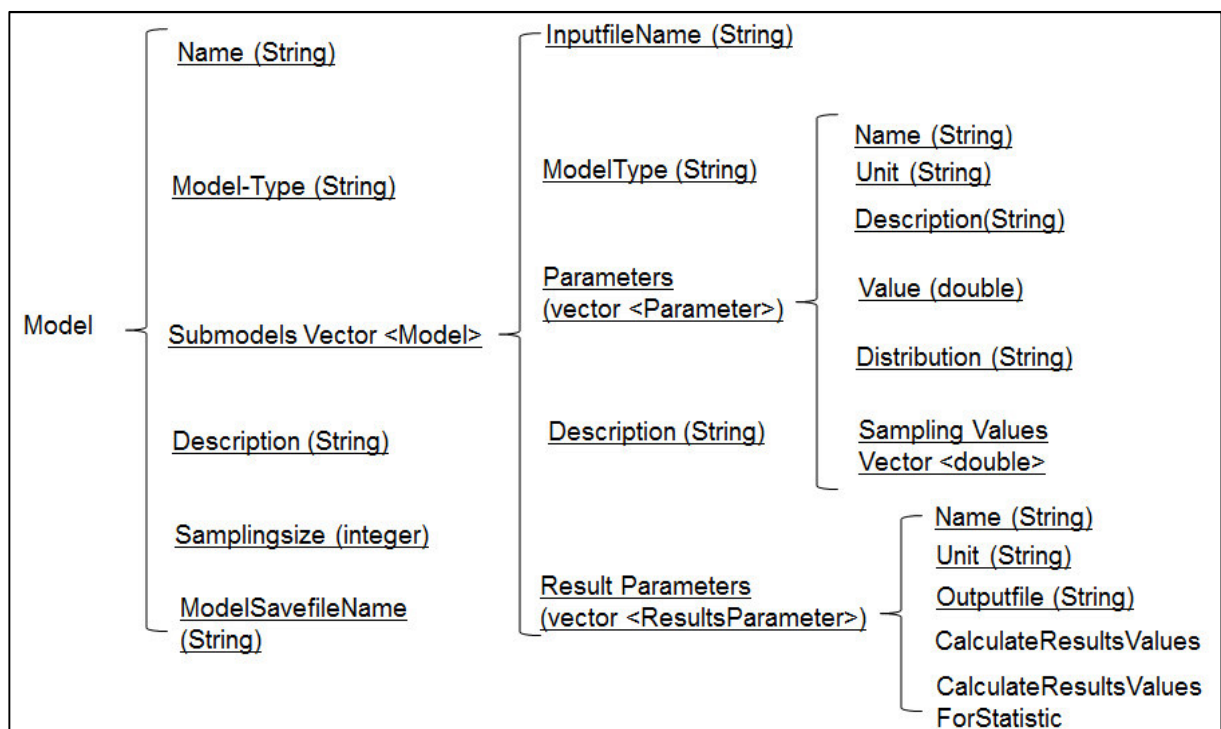


Abb. 5.1 Vereinfachte Klassenstruktur eines Modells in RESUS

Abb. 5.1 zeigt den vereinfachten Klassenbaum, wobei einige wichtige Objekte dargestellt werden. Die durch Unterstreichungen gekennzeichneten Klassen und Objekte werden im Pre-Processing erstellt, die „CalculateResultsValues“ werden während der Simulationsphase beschrieben, in die „CalculateResultsValuesForStatistic“ werden Daten im Post-Processing zur Darstellung und statistische Analyse eingelesen. Die Attribute eines C++ Objekts ist selber auch als Objekt implementiert. Die Zwei Begriffe werden in diesem Kapitel häufig verwendet.

In diesem Kapitel werden die Klassen und dazugehörigen Methoden als Dokumentation für die Weiterentwicklung vorgestellt.

### 5.1.1 Die Klasse „Model“

Der Simulationsprozess in RESUS wird mit dem Objekt „Model“ realisiert. Die Definition und der Bearbeitungscode des Objekts „Model“ befinden sich in den Dateien „model.h“ und „model.cpp“. Beide dienen zur Durchführung der Modellierung. Am Anfang des Pre-Processings wird das Objekt „Model“ initialisiert. In Abb. 5.2 werden die wichtigsten Attribute dargestellt, die später im Lauf der Simulation und Analyse im Objekt „Model“ initialisiert werden. Die meisten dieser Attribute werden als „private“ gekapselt, sind damit von direktem Zugriff von außerhalb der Klassenhierarchie geschützt und sind mit klassischen set- und get-Methoden zum Auslesen und Beschreiben zu erreichen. Die Klasse „Model“ enthält neben den Grundattributen noch andere Objekte als Attribute. Die Schreibweisen unterscheiden sich in Abb. 5.2: die obere Reihe zeigt die Grundattributen „Name“, „Description“, „Parameter“, „ParameterDeterministic“, „Geoformation“ usw. werden in der unteren Reihe als Zusatzattribute dargestellt. Als Grundinformation hat das Objekt „Model“ drei String-Attribute: „Name“, „Description“ und „ModelType“. Das String-Attribut „Name“ enthält den Namen des erstellten Modells. Das String-Attribut „Description“ beinhaltet die Beschreibung des Modells, welche dem Nutzer Information über das Modell liefert. Das dritte String-Attribut „ModelType“ entscheidet, mit welchem Simulationsprogramm das Modell berechnet werden soll. Das Attribut „Timesteps“ ist ein Zahlenvektor, der die Zeitreihe für die Ergebnisdaten festlegt. „Sampling-size“ ist eine Ganzzahl und beschreibt die Größe der Samples für die statistischen Simulationen.

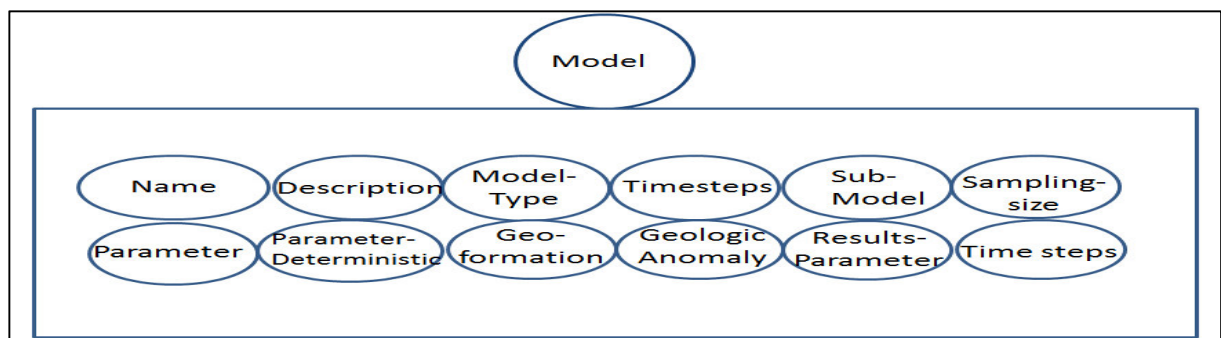


Abb. 5.2 Wichtigste Attribute der Klasse „Model“, in der oberen Reihe die Grundattribute, in der unteren Reihe die Zusatzattribute

Die weiteren Attribute der Klasse „Model“ werden in den folgenden Kapiteln vorgestellt. Die vollständige Auflistung der Objekte, Attribute, und deren Methoden befinden sich in Tabelle 7-1 im Anhang dieser Dissertation.

### 5.1.2 Die Klasse „SubModel“

Die Modellkette in RESUS kann aus unterschiedlichen Modellkomponenten aufgebaut werden, die im RESUS-Quellcode als „Submodel“ bezeichnet werden. Das Objekt „Submodel“ ist ein Attribut des „Model“-Objekts. Die Definition und der Bearbeitungscode für die Submodel-Klasse befinden sich in den Dateien „submodel.h“ und „submodel.cpp“. Die Klasse „Submodel“ wird nur für die Kopplung der Modellkomponenten gebraucht, wenn das Modell mit der dateibasierten Methode erstellt wird. Die Kopplung von Modellkomponenten in RESUS ist, wie schon oben beschrieben, nur mit der dateibasierten Methode machbar.

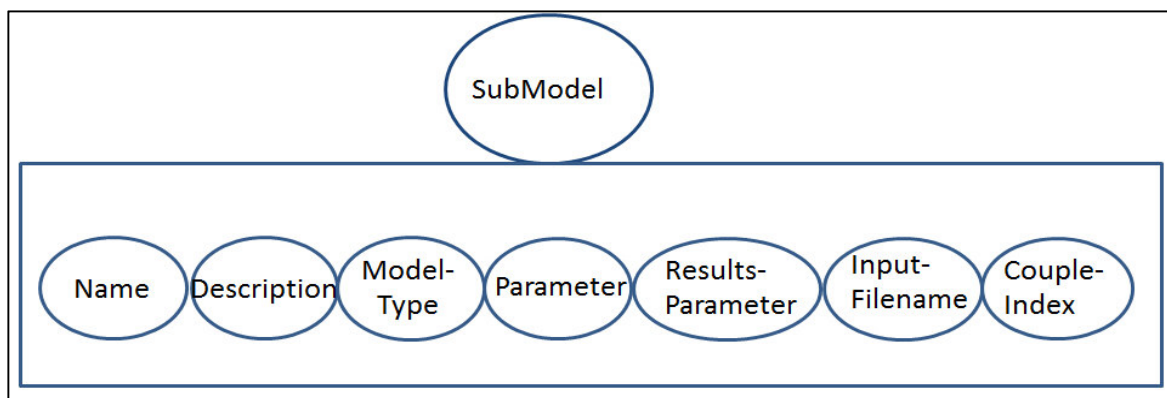


Abb. 5.3 Attribute der Klasse „SubModel“

Abb. 5.3 stellt die Objekte der Klasse „SubModel“ dar. Die Attribute und Objekte dieser Klasse sind größtenteils „private“ und sind mit set- und get-Methoden zu erreichen. Ähnlich wie bei „Model“ hat „SubModel“ die String-Attribute „Name“, „Description“ und „Model-Type“. „Model-Type“ bestimmt das fremde Simulationsprogramm, das für die Berechnung des Submodells genutzt wird. Das Attribut „Parameter“ stellt die probabilistischen Parameter als C++ Vektor dar. Bei der GUI-basierten Methode werden die „Parameter“ als Objekt von „Model“ implementiert, da das Modell nur eine Modellkomponente hat. Bei der dateibasierten Modellerstellungsmethode mit mehreren Submodellen hingegen werden die Objekte „Parameter“ zu „SubModel“ untergeordnet, damit die Modellstruktur klarer aufgebaut wird. „Input-Filename“ beinhaltet die Dateinamen der Template-Eingabedateien in der dateibasierten Methode. „CoupleIndex“ ist eine Ganzzahl, die die Reihenfolge der betrachteten Submodelle in der Modellkette angibt. Im Vergleich mit der Klasse „Model“ braucht die Klasse „SubModel“ kein Objekt „ParameterDeterministic“. Dieses wird bei der GUI-basierten Methode zur Speicherung

der deterministischen Parameter verwendet. Das hier betrachtete „SubModel“ Objekt wird jedoch nur bei der dateibasierten Modellerstellung verwendet. Bei dieser Art der Modellerstellung werden die deterministischen Parameter in den Template-Eingabedateien hinterlegt und müssen nicht in „ParameterDeterministic“-Objekt gespeichert werden. Bei der ersten Initialisierung des „Model“-Objekts werden die Modellkomponenten als „SubModel“-Objekte mitinitialisiert und zu einer Modellkette verknüpft.

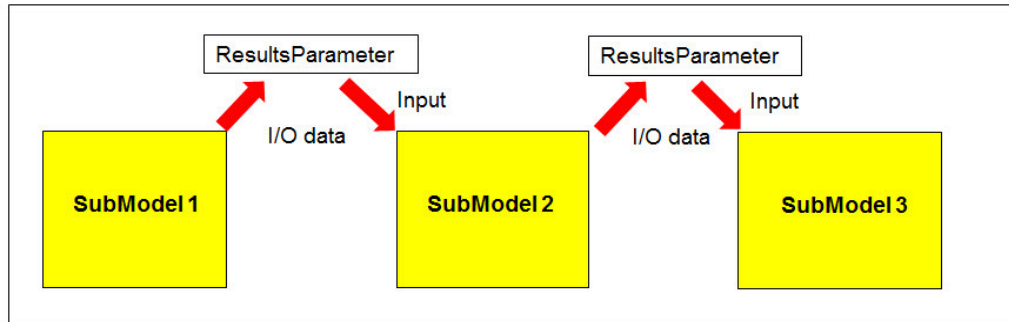


Abb. 5.4 Modellkette eines Simulationsmodells bei RESUS

Abb. 5.4 stellt die Modellkette grafisch dar. Die von einer Modellkomponente errechneten Daten werden zum Objekt „Resultsparameter“ weitergegeben und ebenfalls an das nächste „SubModel“ als zeitabhängige Eingabedaten (z.B. Randbedingung, Quellterm, zeitabhängige Materialparameter usw.) weitergeleitet. Die resultierenden Ergebnisse werden danach gespeichert.

### 5.1.3 Die Klasse „Parameter“

Bei der Initialisierung eines Modells und seiner Submodelle werden die dazugehörigen Parameter als C++ Vektor mit Parameter-Objektreihe benötigt. Die Klasse „Parameter“ beschreibt die probabilistischen Parameter des „Model“ Objekts. Sie ist sowohl im Modell mit Submodellen selbst als auch in denjenigen Submodellen zu finden, für die eine probabilistische Analyse mit statistischen Samplingwerten durchgeführt werden soll. Abb. 5.5 stellt die Attribute von „Parameter“ dar. Die Definition und der Bearbeitungscode der Klasse „Parameter“ befindet sich in den Dateien „parameter.h“ und „parameter.cpp“.

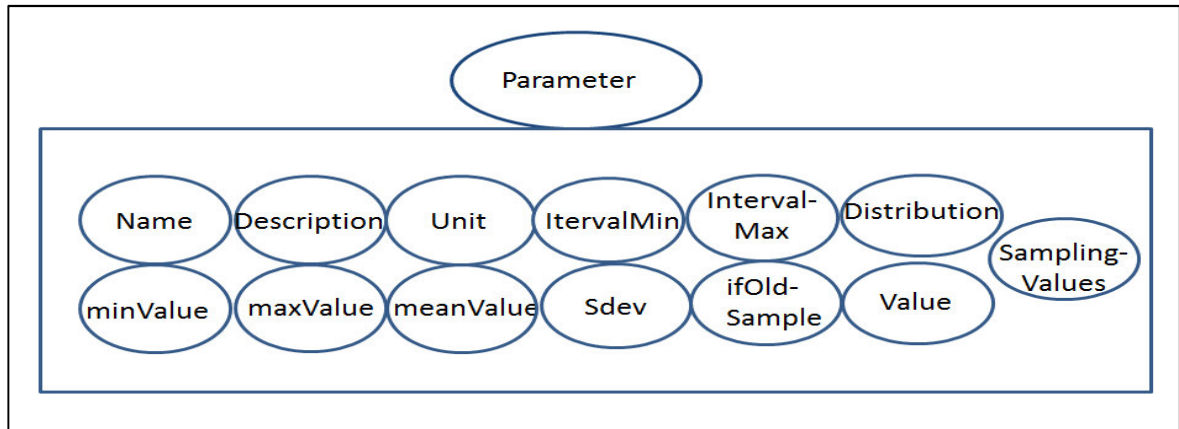


Abb. 5.5 Attribute der Klasse „Parameter“

Die Objekte „Name“, „Description“ und „Unit“ bezeichnen die Grundinformation des Parameters. „IntervalMin“ und „IntervalMax“ begrenzen den physikalischen Wertebereich des Parameters. Alle Zufallswerte für diesen Parameter liegen in diesem Bereich. „Distribution“ beinhaltet die festgelegte statistische Verteilung des Parameters. „Distribution“ ist ein String-Objekt, das den Namen der statistischen Verteilung des Parameters enthält. Der Verteilungsname entscheidet, wie die Zufallswerte durch den Zufallsgenerator erzeugt werden sollen. Wenn die statistische Verteilung eines Parameters eine „Gleichverteilung“ oder „logarithmische Gleichverteilung“ ist, braucht der Parameter die zusätzlichen Attribute „minValue“ und „maxValue“ für den Zufallsgenerator. Ebenso benötigt ein nach „Gaußverteilung“ oder „logarithmischer Gaußverteilung“ spezifizierter Parameter die Attribute: Mittelwert („meanValue“) und Standardabweichung („Sdev“). Die „SamplingValues“ sind als C++ Vektor mit „double“ als Datentyp festgelegt. In diesem Vektor werden die gesampelten Werte für die Parameter gespeichert. „Value“ ist eine Zahl des Datentyps „double“, die den Referenzwert des probabilistischen Parameters angibt. Die alle im Text genannten „Vektor“ sind Vektor der C++ Programmiersprache, sie haben keine physikalische Bedeutung.

Wenn RESUS ein schon vorher errechnetes Modell aus gespeicherten Dateien einliest, kann der Nutzer für einen Parameter bei der Sampleerzeugung die Option „Use the old samples“ auswählen, die in Kapitel 4.3.2 mit einem Beispiel genauer erklärt ist. Die getroffene Wahl wird in eine boolesche Variable „ifOldSample“ gespeichert. Ist „ifOldSample“ auf „true“ gesetzt, werden die Werte nicht vom Zufallsgenerator neu gesampelt, sondern aus den vorher gespeicherten Ergebnisdateien eingelesen. Die vollständigen Attribute des Objekts „Parameter“ werden in Tabelle 7-3 im Anhang aufgeführt.



#### 5.1.4

#### Die Klasse „ResultsParameter“

Die Klasse „ResultsParameter“ bezeichnet die Parameter, die von der Simulation ausgegeben werden. Bis auf die Grundattribute „Name“, „Unit“, „Description“ usw. unterscheiden sich die Ergebnisparameter in ihren Attributen nicht von den Eingabeparametern. Die Attribute und Methoden der Klasse „ResultsParameter“ befinden sich in den Dateien „resultsParameter.h“ und „resultsParameter.cpp“.

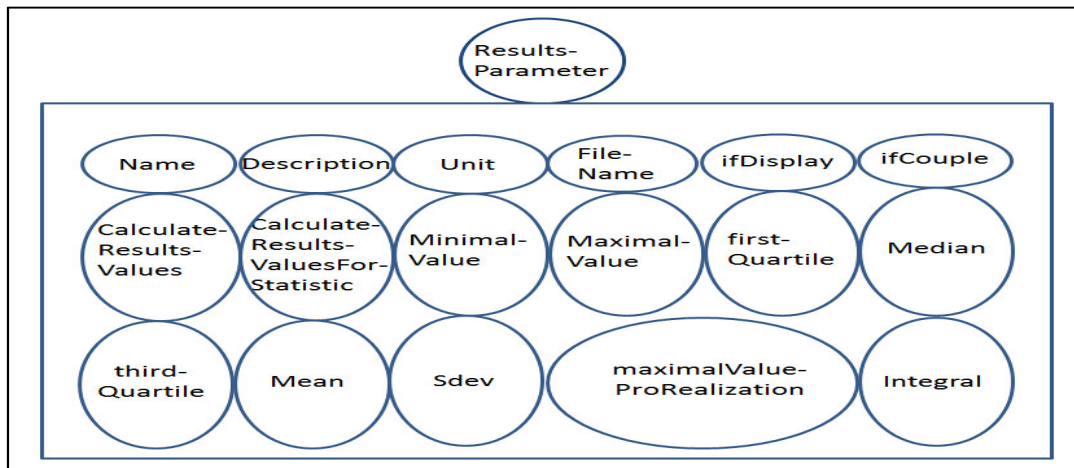


Abb. 5.6 Attribute der Klasse „ResultsParameter“

RESUS ist eine Simulationsplattform, in der numerische Berechnungen durch externe Fremdcodes durchgeführt werden können. In einem Ausgabeparameter werden die errechneten Werte gespeichert, die aus einer Ausgabedatei des Fremdcodes eingelesen wurden. Das Objekt „Resultsparameter“ muss ein Attribut „FileName“ für die Kennzeichnung der Datei haben, aus der die eingelesenen Werte stammen. Das boolesche Attribut „ifDisplay“ enthält die Information, ob der Benutzer den Ausgabeparameter beim Post-Processing der Simulation darstellen und analysieren lassen möchte. Die Simulationsergebnisse bei Endlagerfragestellungen sind normalerweise zeitabhängig. Aus diesem Grund sind die Werte der meisten Parameter zweidimensional. Deswegen ist der Vektor „calculateResultsValues“ (Vector<Vector<double>>), in dem die Rechenergebnisse gespeichert werden, zweidimensional. Die erste Dimension wird für die zeitabhängigen Ergebniswerte benötigt, die zweite Dimension für die Werte aus unterschiedlichen Rechenrealisierungen. Das Attribut „calculateResultsValueForStatistic“ dient der statistischen Verarbeitung der eingelesenen Ergebnisdaten. Die errechneten Werte in „calculateResultsValues“ werden zunächst in den „calculateResultsValueForStatistic“ kopiert, dann werden die Werte zu jedem fixen Zeitpunkt der Messung der Größe nachgeordnet. Dies dient zur Vorbereitung der späteren statistischen Auswertung.

Der eindimensionale Vektor „minimalValue“ speichert die minimalen Werte jedes Simulationszeitschrittes aus dem zweidimensionalen Vektor „calculateResultsValueForStatistic“. Diese verbinden sich zu einer Wertkurve in der Zeitdimension, die als 0%-Quantil für die



Analyse und Darstellung genutzt wird. Ähnlich wie bei „minimalValue“ gibt es noch weitere eindimensionale Vektoren, welche die statistischen Quantile der Rechenrealisierungen enthalten: „firstQuartile“ für das 25%-Quantil, „median“ für das 50%-Quantil, „thirdQuartile“ für das 75%-Quantil und „maximalValue“ für die höchsten Werte aller Zeitschritte (100%-Quantil). Alle diese statistischen Maße sind eindimensionale Vektoren, welche ihre Werte aus dem „calulaterResultsValueForStatistic“-Vektor beziehen.

Die Vektoren „Mean“ und „Sdev“ beinhalten die statistischen Maße: Arithmetischer Mittelwert und Standardabweichung der Rechenergebnisse. Die Werte aller Realisierungen zu jeden Zeitschritt aus dem „calulaterResultsValueForStatistic“-Vektor werden erneut zur Berechnung genutzt (siehe Gleichung 5-1, 5-2).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5-1)$$

$$\text{Var}(X) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5-2)$$

dabei beschreibt  $x_i$  alle Werte der Rechenrealisierungen an dem fixen Zeitschritt,  $n$  steht für die Anzahl der Realisierungen.  $\bar{x}$  bezeichnet den Mittelwert der Datenreihe.

Die Methoden für die Berechnung der statistischen Maße befinden sich in der Klasse „Datacondition“, und werden in Kapitel 5.3.6 genauer beschrieben.

Die meisten Ausgabeparameter der Rechenrealisierung in RESUS sind zeitabhängig, enthalten also Verläufe des Parameterwerts über die betrachtete Simulationszeit. RESUS ermittelt jedoch keine zeitabhängigen Sensitivitätsmaße. Die in jedem Durchlauf entstandenen Daten müssen daher auf nur einen Wert reduziert werden. Dafür ist der oben schon erwähnte „Quantity of interest“ zuständig. Unter dem „Quantity of interest“ verstehen wir eine mathematische Rechenmethode, die vom Benutzer ausgewählt werden kann. Sie reduziert die zeitabhängigen Daten auf einen skalaren Wert.

In der Version 1.0 von RESUS werden zwei „Quantity of interest“ angeboten: Zum einen der maximale Wert jeder Realisierung und zum anderen das Integral der Rechenwerte über die Zeit. Beide wurden als Attribute des Objekts „Resultsparameter“ mit den Namen „maxPerRealization“ und „Integral“ programmiert. Diese „Quantity of interest“ verarbeiten die zweidimensionalen Ergebnisse zunächst zu einem eindimensionalen Vektor mit dem Datentyp <double>. Die Berechnung der zwei „Quantity of interest“ erfolgt für den Maximalwert nach Gleichung 5-3 und für die Integration nach Gleichung 5-4:

$$\text{Max\_pre\_realization} = \text{Max}_{t=0}^{\text{time length}} x_t \quad (5-3)$$

$$\text{Integral\_per\_realization} = \sum_{t=0}^{\text{time length}} (x_t + x_{t-1}) * \Delta t / 2 \quad (5-4)$$

Gleichung 5-3 gibt an, dass der größte Wert aller in der Simulationszeit berechneten Werte aus den Ergebniswerten herausgegriffen wird. Das Ergebnis dieser Berechnung wird im eindimensionalen Vektor „maxPerRealization“ gespeichert.

Die Gleichung 5-4 beschreibt das Berechnungsverfahren für den „Quantity of interest“: „Integral“. Dieser integriert den Verlauf eines jeden Rechendurchlaufs, indem die zeitabhängigen Ergebnisse mit den Zeitschrittlängen multipliziert und dann addiert werden. Das Ergebnis aus jedem Durchlauf wird im eindimensionalen Vektor „Integral“ gespeichert. „Integral“ ist dafür geeignet, die Ausgabegröße der Nuklidkonzentration über die Zeit zu summieren. Damit erhält man folglich die Gesamtmenge der Radionuklidfreisetzung im Zeitraum der Analyse.

Die vollständigen Attribute des Objekts „ResultsParameter“ werden in Tabelle 7-4 im Anhang aufgeführt.

## **5.2                   Andere Klassen für den Modellaufbau**

Die in Kapitel 5.1 vorgestellten Klassen werden für alle Simulationsanalysen mit RESUS gebraucht. Wegen der zwei Modellkonstruktionsmethoden (GUI-basierte Methode und dateibasierte Methode) wurden einige Klassen spezifisch für bestimmte Modelle und bestimmte Modellaufbauverfahren implementiert. Diese werden in diesem Kapitel beschrieben.

### **5.2.1               Die Klasse „ParameterDeterministic“**

Die Klasse „ParameterDeterministic“ wird für die Parameter gebraucht, die in der Simulation als deterministische Parameter behandelt werden müssen. Sie wird ausschließlich bei der Modell-erstellung mit der GUI-basierten Methode genutzt, da die deterministischen Parameter bei der dateibasierten Methode in der Template-Eingabedatei direkt einzugeben sind. Für das zweidimensionale Transportmodell mit der GUI-basierten Modellerstellungsmethode wird diese Klasse gebraucht. Beim eindimensionalen Transportmodell werden alle Parameter im Hinblick auf den Programmieraufwand als probabilistische Parameter betrachtet, da es relativ wenige Parameter gibt, die konstant gehalten werden müssen. Die Klasse „ParameterDeterministic“ findet man in den Dateien „parameterDeterministic.h“ und „parameterDeterministic.cpp“.

Die Klasse beginnt mit einem Konstruktor, um das Objekt „ParameterDeterministic“ selbst und dessen Attribute zu initialisieren. „ParameterDeterministic“ besitzt auch die Grundinformation wie „Name“, „Unit“ und „Description“.

Anders als bei den probabilistischen Parametern brauchen die deterministischen Parameter nur konstante Werte, die später zur Simulation immer gleichbleibend einzusetzen sind. Mit den

Methoden setValue() und getValue() kann auf die deterministischen Werte eines Parameters zugegriffen und dieser geändert werden.

Die Tabelle für die Auflistung der vollständigen Attribute der Klasse „ParameterDeterministic“ befindet sich in Tabelle 7-5 im Anhang.

### 5.2.2 Die Klasse „Geoformation“

Die Klasse „Geoformation“ ist für ein Stofftransportmodell in zweidimensional modellierten geologischen Formationen gedacht. Ein solches Stofftransportmodell besteht aus geologischen Schichten (geologischen Formationen) und wird mit der GUI-basierten Methode aufgebaut. Jede im Modell betrachtete geologische Formation wird in RESUS-Quellcode als Objekt „Geoformation“ bezeichnet.

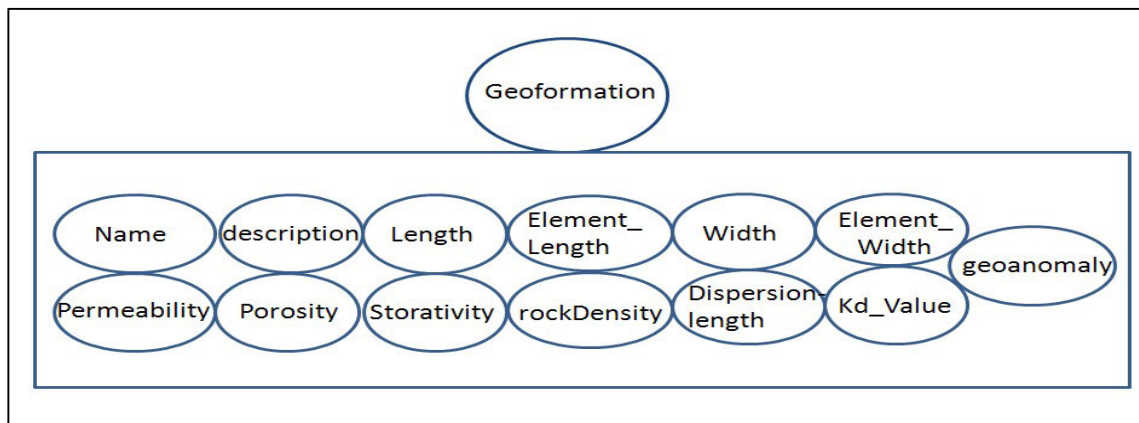


Abb. 5.7 Attribute der Klasse „Geoformation“

Die Klasse „Geoformation“ beschreibt eine geologische Formation, die als Transportmedium dient (siehe Abb. 5.8). Das Modell wird so eingestellt, dass die Schadstoffe durch Wasser in den Geomedien transportiert werden. Jede Geoformation hat eigene geohydraulische Eigenschaften („Porosity“, „Permeability“ usw.), die im Transportmodell als probabilistische Parameter berücksichtigt sind und in der Klasse als Attribute hinterlegt werden. Auf die Parameter wird mit Methoden der Klasse „set()“, und „get()“ zugegriffen. Die Geoformationen sind im Objekt „model“ immer als Vector<geoformation> zu implementieren. Die Geoformation werden geometrisch nach ihrer Reihenfolge in den Vektor zugeordnet (siehe Abb. 5.8). Die zweidimensionale Geometrie (Länge und Breite) und die Meshing-Methode (Element\_length und Element\_Width) für die FEM-Simulation bestimmen den Gitteraufbau des Simulationsmodells. Die Gitteraufbaumethode wird in Kapitel 5.3.3 beschrieben.

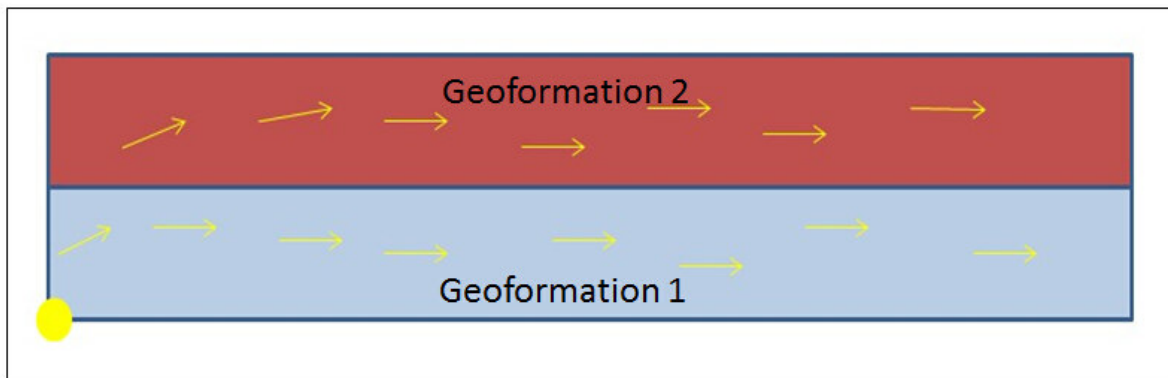


Abb. 5.8 Transport in zwei geologischen Formationen

In Abb. 5.8 strömt Grundwasser von links nach rechts, dabei wird Schadstoff transportiert. Der gelbe Punkt links unten ist die Quelle der Schadstofffreisetzung, die das Wasser im gesamten Gebiet kontaminiert. Geologische Formationen haben in der Regel einen ungleichmäßigen geologischen Aufbau. Sie können bereichsweise aus unterschiedlichen Materialien mit unterschiedlichen Eigenschaften bestehen, wie in Abb. 5.9 gezeigt. Die rote geologische Schicht hat eine „Lücke“ (hellblauer Bereich) und einen dunkelblauen Einschluss. Diese Störungen haben andere hydraulische Transportmechanismen als der umgebende Bereich. Beide Bereiche werden als „GeologicAnomaly“ gezeichnet und in RESUS als eigene Klasse definiert. Das Objekt „Geoformation“ kann eine Reihe von GeologicAnomaly haben, die im Vector<GeologicAnomaly> gespeichert werden. Die „GeologicAnomaly“ als Klasse wird in Kapitel 5.2.3 beschrieben.

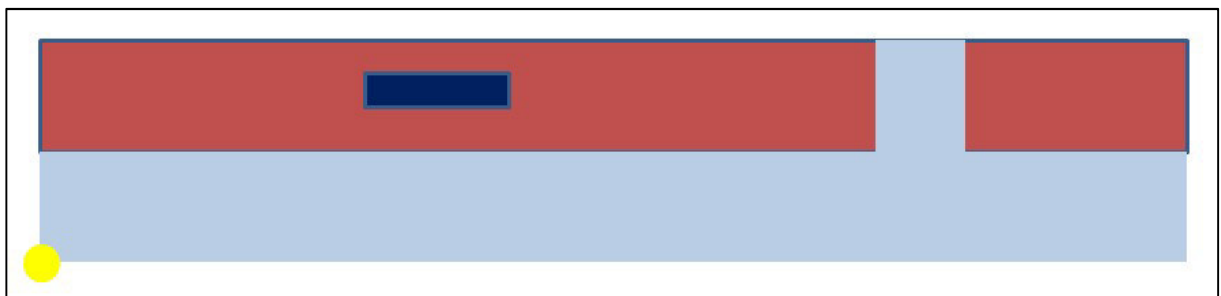


Abb. 5.9 Zwei geologische Formationen mit unterschiedlichen Transporteigenschaften. Die rote Schicht hat zwei Anomalien: die dunkelblaue Zone und die hellblaue Zone

### 5.2.3 Die Klasse „GeologicAnomaly“

Die Klasse zur Modellierung der Geoanomalien ist ein Bestandteil des in Kapitel 4.4 vorgestellten Transportmodells. Sie wird als „GeologicAnomaly“-Objekt in die das verwendete „Geoformation“-Objekt eingefügt. Die Attribute von „GeologicAnomaly“ finden sich in Abb. 5.10. Die Klasse wird in den Dateien „geologicanomaly.h“ und „geologicanomaly.cpp“ definiert und implementiert.

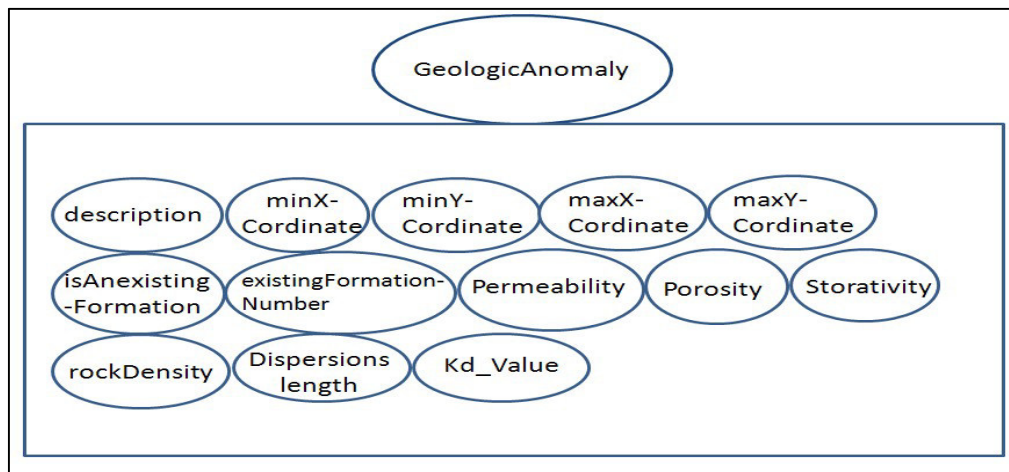


Abb. 5.10 Objekte und Methoden der Klasse „GeologicAnomaly“

Die Abb. 5.20 zeigt die Attribute der Klasse „Geologicanomaly“. Die Geoanomalien sind als C++ `Vector<GeologicAnomaly>` gruppewise in die „Geoformation“ eingefügt.

Um den Modellaufbau zu vereinfachen, werden die Geoanomalien immer in rechteckiger Form dargestellt. Der rechteckige Bereich muss vier Koordinatendaten (minimale x-Cordinate, maximale x-Cordinate, minimale y-Cordinate und maximale y-Cordinate) haben, um die Position und Geometrie der Anomalie festzulegen. Die hydraulische Eigenschaft der Geoanomalie kann vom Benutzer auf zwei verschiedene Weisen festgelegt werden (siehe Abb. 5.11):

1. Wie im Bereich 2 der Abb. 5.11 dargestellt kann der Nutzer den gleichen Gesteinstyp wie in einer anderen geologischen Formation angeben. In diesem Fall wird die Variable der Anomalie „isAnexistingFormation“ zu einer bestimmten Geoformationsnummer gesetzt und deren Transporteigenschaften werden direkt von der anderen Formation übernommen. Z.B. wird für die Anomalie 2 im Beispiel die „isAnexistingFormation“ auf „1“ gesetzt, damit sie die identische Eigenschaft wie Geoformation 1 hat.
2. Die zweite Möglichkeit ist wie im Anomalie 1 in Abb. 5.11 dargestellt: Der Anomaliebereich hat einen eigenen Gesteinstyp, deswegen wird „isAnexistingFormation“ von Anomalie 1 auf „false“ gesetzt. Die entsprechenden hydraulischen Parameter (Porosität, Permeabilität,  $K_d$ -Wert usw.) müssen zusätzlich von Benutzer angegeben werden.

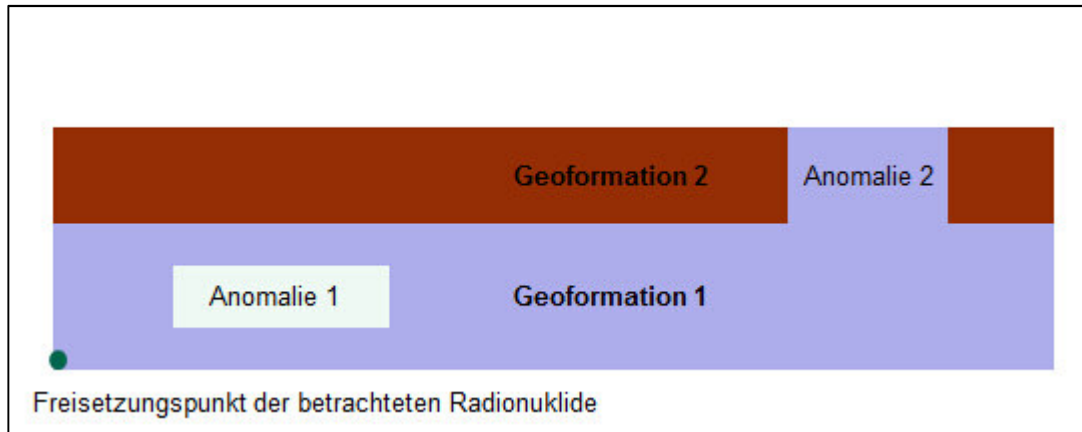


Abb. 5.11 Zwei geologische Formationen mit unterschiedlichen Transporteigenschaften. Die rote Schicht hat zwei Anomalien: die dunkelblaue Zone und hellblaue Zone

Mit den Klassen „Geoformation“ und „geologicAnomaly“ kann ein zweidimensionales Schadstoff-Transportmodell als Kontaminationsverteilungsbeispiel (siehe Kapitel 4.5) mit der GUI-basierten Methode aufgebaut werden. Die vollständige Darstellung der Attribute der Klasse „GeologicAnomaly“ befindet sich in Tabelle 7-7 im Anhang.

### 5.3 Anderen Bestandteile in RESUS

RESUS hat eine objektorientierte Basisstruktur. Wie vorher beschrieben, werden die Modelle in RESUS nicht mit einem eigenen Solver berechnet, sondern mit Hilfe externer Fremdcodes berechnet. Für jede Implementierung eines Fremdcodes muss in RESUS eine Schnittstelle geschaffen worden sein, die die Kommunikation zwischen RESUS und den Fremdcodes ermöglicht. Für die Implementierung unterschiedlicher Fremdcodes werden die Schnittstellen in RESUS in Quelldateien eingebunden. Jede dieser Dateien enthält generell eine Set-Methode für die Initialisierung des Modells, eine Dateierstellungsmethode (File-Set-Methode) für die Erzeugung der Inputdatei des Fremdcodes, eine Kalkulationsmethode für den Aufruf des Fremdcodes, sowie eine Ergebnislesemethode für das Lesen der Rechenergebnisse aus dem Fremdcode. Diese Struktur und der Arbeitsablauf werden für die Implementierung der Fremdcodes eingesetzt und in diesem Kapitel vorgestellt.

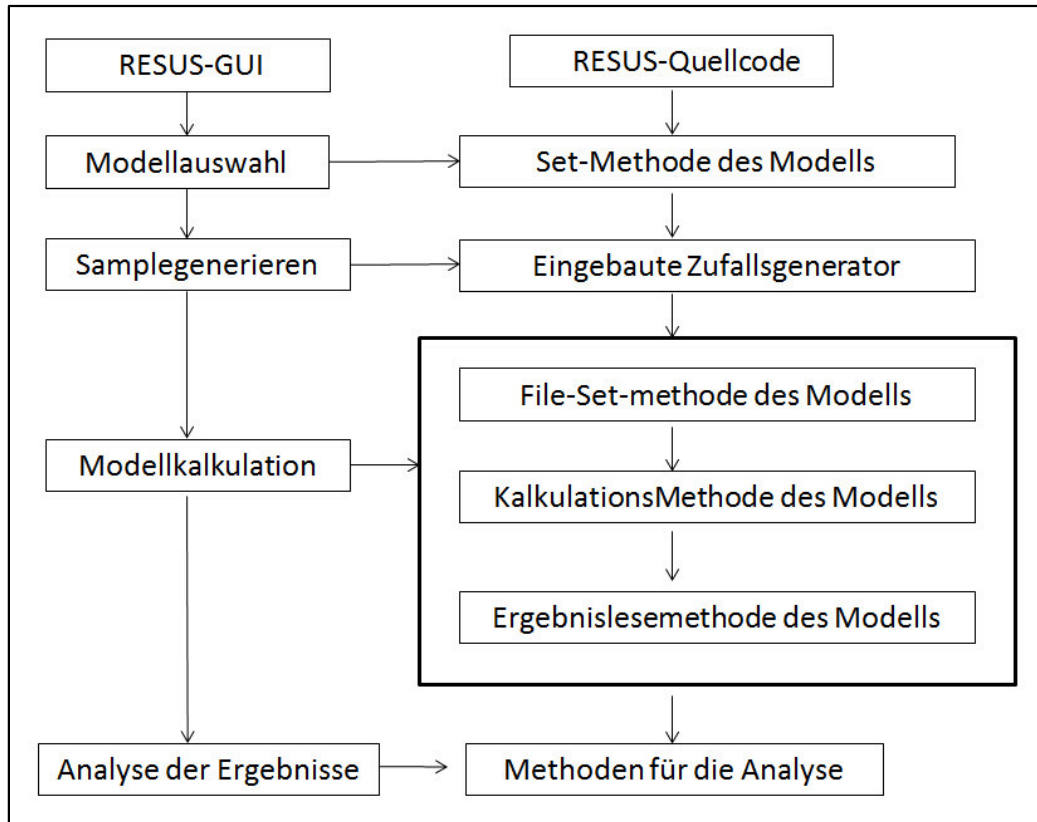


Abb. 5.12 Arbeitsprozedur von RESUS: links ist die Kommando von Nutzer auf der GUI, links ist die entsprechenden Prozesse, die RESUS Quellcode ausführt

Neben den Interfaces zu Fremdcodes werden auch die Zufallsgeneratoren und die Speicher- und Wiederöffnungsmethoden in RESUS in diesem Kapitel dargestellt.

### 5.3.1 Zufallsgenerator

Für die statistische Analyse wird ein Zufallsgenerator gebraucht. Der Zufallsgenerator in RESUS erzeugt die zufälligen Werte mit der „Simple Random Sampling“-Methode (SRS). Der Zufallsgenerator der aktuellen Version von RESUS kann Zufallswerte für fünf Typen von Zufallsverteilungen erstellen: Gleichverteilung, Gaußverteilung, logarithmische Gleichverteilung, logarithmische Gaußverteilung, Dreiecks-Verteilung.

Der Zufallsgenerator ist von der RESUS getrennt als externe Methode eingebaut. Die mathematische open Source Code-Bibliothek StochasticLib [59] wird für dessen Erstellung genutzt. Der Zufallsgenerator wird auf RESUS aus den Dateien „stoc1.h“, „stoc1.cpp“ und „randomc.h“ „randomc.cpp“ kompiliert. Die Methoden in diesen vier Dateien erzeugen

Zufallszahlen für eine große Anzahl von Verteilungen. In RESUS werden lediglich die Methoden für die Gleichverteilung und die Gaußverteilung genutzt.

Die gleichverteilten oder normalverteilten Zufallswerte werden mit Hilfe der Zufallsgeneratoren in „StochasticLib“ erzeugt. Die erzeugten Samples werden im Vektor „samplingValues“ mit dem Datentyp <double> ins Objekt „Parameter“ eingefügt. Für die Sampleerzeugung nach der logarithmischen Gleichverteilung, logarithmischen Gaußverteilung und Dreiecks-Verteilung erfolgt in RESUS mit Hilfe der bestimmten mathematischen Transformationen.

$$\text{samplewert} = \exp(\text{generatorUniform}(\log(\min), \log(\max))) \quad (5-5)$$

Gleichung 5-5 zeigt, dass die Verteilungsmaße (minimale und maximale Wert) für die logarithmische Gleichverteilung zuerst logarithmisch transformiert werden. Mit den transformierten Werten werden gleichverteilte Zufallszahlen erzeugt, die dann wieder exponentiell transformiert und zum Vektor „SamplingValues“ im „Parameters“-Objekt hinzugefügt werden. Die transformierten Werte sind dann logarithmisch gleichverteilt.

$$\text{samplewert} = \exp(\text{generatorNormal}(\log(\text{mean}), \text{stdv})) \quad (5-6)$$

Der Zufallsgenerator erzeugt die Zufallszahlen mit dem logarithmischen Mittelwert und der Standardabweichung nach Gaußverteilung, die Werte werden dann exponentiell transformiert und in den Vektor „samplingValues“ des „parameters“-Objekts eingefügt. Die transformierten Werte verteilen sich dann nach der logarithmischen Gaußverteilung.

Für die Gaußverteilung und logarithmische Gaußverteilung bietet RESUS noch eine Funktion namens „truncate“ an. Eine „while“-Schleife wird für diese Funktionalität verwendet. Wenn der Benutzer die Zufallsproben mit „truncate“ erzeugen lässt, wird nach der Erzeugung der Zufallszahlen kontrolliert, ob die Werte innerhalb eines vorher festgelegten Intervalls liegen. Ist dies nicht der Fall, wird der Erzeugungsprozess erneut durchgeführt, bis ein geeigneter Wert erzeugt ist.

Die Dreiecksverteilung hat neben maximalen und minimalen Wert noch einen „middle“ als Modalwert mit maximaler Wahrscheinlichkeitsdichte. Zufallswerte, die nach der Dreiecksverteilung verteilt sind, können durch den Zufallsgenerator für die Gleichverteilung mit folgender Transformation erzeugt werden:

$$\text{sample} = \text{generatorUniform}(0, 1.0) \quad (5-7)$$

$$\text{sample\_tri} = \min\text{Value} + \sqrt{\text{sample} \times (\text{middle} - \min\text{Value}) \times (\max\text{Value} - \min\text{Value})} \quad (5-8)$$

$$\text{sample\_tri} = \max\text{Value} - \sqrt{(1 - \text{sample}) \times (\text{middle} - \min\text{Value}) \times (\max\text{Value} - \min\text{Value})} \quad (5-9)$$



RESUS erzeugt zuerst einen gleichverteilten Wert „sample“ zwischen 0 bis 1. Wenn der erzeugte Wert „sample“ kleiner als  $(middle - minValue) / (maxValue - minValue)$  ist, wird er nach Gleichung 5-8 zu „sample\_tri“ transformiert. Sonst wird er nach Gleichung 5-9 transformiert. Die transformierten Werte „sample\_tri“ sind dann dreiecksverteilt und werden im Vektor „samplingValues“ des „parameters“-Objekts eingefügt.

### 5.3.2 Ishigami Modell

Das Ishigami Modell wird als Testmodell in RESUS mit der GUI-basierten Methode aufgebaut. Die Methoden für das Modell werden in den Dateien „ishigami.h“ und „ishigami.cpp“ implementiert. Beim Aufruf der Auswahl „Ishigami“ Modell wird die Methode „ishigamiSet(\*model)“ aufgerufen. Mit dieser Methode wird die Grundinformation („Name“, „ModelType“ und „Description“) des Objekts „Model“ hinterlegt. Die vier probabilistischen Parameter ( $X_1$ ,  $X_2$ ,  $X_3$  und  $X_4$ ) werden mit dem Konstruktor der Klasse „Parameter“ nacheinander in den Vector<Parameter> ins Objekt „Model“ eingefügt. Der Ausgabeparameter Y wird einschließlich in den Vektor „ResultsParameter“ des „model“s implementiert.

Nach der Ausführung der Samplegenerierung mit Hilfe des eingebauten Zufallsgenerators kann das Modell berechnet werden. Da das Ishigami-Modell nur eine Gleichung hat, wird die Modellgleichung direkt in die Methode „ishigamiCalculate(model)“ implementiert.

Mit dem Aufruf der Methode wird das Modell berechnet. Wenn die Simulationen abgeschlossen sind, wird das Ergebnis mit Hilfe der Methode „loadResultsOfIshigami(\*model)“ zu „ResultsParameter“ eingelesen. Da das Ishigami-Modell ein zeitunabhängiges Ergebnis hat, hat der Vektor „ResultsParameter“ in diesem Fall nur einen Wert je Realisierung zu speichern.

### 5.3.3 Level-E Modell

Das Level-E Modell wird in RESUS mit der GUI-basierten Methode aufgebaut. Die Methoden des Modells werden in den Dateien „levele.h“ und „levele.cpp“ implementiert.

Wenn der Benutzer in der RESUS-GUI das Modell „Level-E“ gewählt hat, wird die Methode „leveleSet(\*model)“ aufgerufen. In dieser Methode wird die Grundinformation („Name“, „ModelType“ und „Description“) des Objekts „Model“ hinterlegt. Die zwölf probabilistischen Parameter werden mit dem Konstruktor der Klasse „Parameter“ nacheinander in den Vector <Parameter> des Objekts „model“ eingefügt. Mit dem Konstruktor werden die Attribute der

einzelnen Parameter initialisiert. Die Ausgabeparameter werden ebenfalls als (Vector<ResultsParameter>) konstruiert.

Nach der Ausführung der Methode „*leveleSet(\*model)*“ ist die Grundstruktur des „model“-Objekts vollständig, einschließlich der Eingabe- und Ausgabeparameter sowie der Zeitschritte. Der Zufallsgenerator kann dann entsprechend der für die probabilistischen Parameter angegebenen Verteilungen Zufallswerte erzeugen und diese zum Vektor „samplingValues“ hinzufügen. Wenn das Sample vollständig ist, kann die Modellkalkulation des Level-E Modells stattfinden.

Im Simulationsprozess schreibt RESUS zuerst die Eingabedateien mit der Methode „*leveleInput(\*model)*“. Diese Eingabedatei für Level-E ist eine textförmige Datei namens „input.sam“, die man mit dem „ofstream“ Objekt editieren kann.

```
0

2 ----- Anzahl der Realisierungen

12 ----- Anzahl der probabilistischen Eingabeparameter

0

300 0.003 3e-006 0.05 200 3 5 0.03 100 3 10 1e+006 --- Samples der 12 Parameter

570.886 0.004 7.7e-006 0.0022 485.16 2.049 20.57 0.0135 193.195 2.97 20.4
9.82911e+006 --- Samples der 12 Parameter der zweiten Realisierung

...
```

Abb. 5.13 Dateiformat der Eingabedatei „input.sam“ vom Level-E Modell

Die Datei für Level-E hat das Format wie in Abb. 5.13. Die in der zweiten Zeile angegebene Zahl (hier „2“) bestimmt die Anzahl der Realisierungen und wird in der GUI wie auch im Quellcode als „SamplingSize“ (Samplegröße) bezeichnet. Die Zahl in der folgenden Zeile bestimmt die Anzahl der probabilistischen Parameter. Die probabilistischen Parameterwerte werden in der fünften Zeile der Eingabedatei aufgeführt. Sie sind genau nach der Reihenfolge in Tabelle 4-1 geordnet. In den nächsten Zeilen werden die Werte für die nachfolgenden Realisierungen in gleicher Reihenfolge aufgeführt.

Die Methode *leveleInput()* in RESUS schreibt zuerst die Eingabedateien für Level-E und hinterlegt darin die gesampelten Parameter aus dem „samplingValue“-Vektor. Ist das geschehen, ruft RESUS die Level-E Applikation „*gtmle.exe*“ mit Hilfe der „*runLevele()*“ Methode auf. Level-E Anwendung liest dann die Eingabedatei ein und führt die Simulation durch. Das Programm „*gtmle.exe*“ erzeugt für das Nahfeld, das Fernfeld und die Biosphärenmodellteile die

Dateien: „*nearf.dat*“, „*geosph1.dat*“, „*geosph2.dat*“ und „*output.dat*“ als Ausgabedateien. Alle der Ausgabedateien haben ein ähnliches Format.

```

RUN      0

      32

1300.000000 2.3360608E-010 1.8710130E-034 1.1589681E-022 5.6274336E-026

1800.000000 2.5147722E-008 5.6117288E-034 3.4712871E-022 1.3924093E-025

2400.000000 1.7572814E-006 2.2344805E-033 1.3788581E-021 5.3314775E-025

3200.000000 2.7614917E-005 1.7434065E-032 1.0585026E-020 4.5891427E-024

...

```

Abb. 5.14 Ausgabedatei „Geosph1.dat“ von Level-E

Abb. 5.14 zeigt ein Beispiel einer Ausgabedatei von Level-E. Die Zahl „32“ steht für die Anzahl der Zeitschritte. Die erste Spalte ist die Zeitreihe, die zweite Spalte ist der Konzentrationswert von I-129, in der dritten bis fünften Spalte sind die Konzentrationen der Np-Zerfallskette, also für die Nuklide Np-237, U-233 und Th-229 aufgeführt.

Wenn die Simulation abgeschlossen ist, ruft RESUS die Methode `loadResultsOfLevelE(*model)` auf. Sofern die Attribute „ifDisplay“ vom „resultsParameter“ auf „true“ gesetzt sind, sucht die Methode deren Ausgabedatei mit Hilfe des Attributes „FileName“ und liest die Daten mit „ofstream“ von der Datei ein. Die eingelesenen Daten werden in den Vektor „calculateResultsValues“ eingeordnet. Ist „ifDisplay“ eines „ResultsParameter“ auf „false“, werden keine weiteren Werte bei ihm eingelesen.

### 5.3.4 Das Transportmodell mit „RockFlow“ und „OGS“

In RESUS werden zur Simulation der Transportprozesse im Deckgebirge das numerische Programm „RockFlow“ und seine Nachfolgeversion „OpenGeoSystem“ (OGS) benutzt.

Für die in diesem Modell integrierbare Adsorptionsisotherme ( $K_d$ -Isotherme) bietet RESUS über die GUI-basierten Methode sechs mögliche Adsorptionsisotherme (siehe Tabelle 4-3) zur Auswahl an. Diese Auswahl wird mit Hilfe des „Switch()-Statement“ implementiert.

Mit der GUI-basierten Methode werden die eindimensionalen und zweidimensionalen Transportmodelle in RESUS erstellt. Alle für die eindimensionalen Transportmodelle verwendeten Methoden zur Initialisierung, Berechnung und für die Einbindung der Fremdcodes befinden sich in den Dateien „geospheremodel1.h“, „geospheremodel1.cpp“. Die Methoden für die zweidimensionalen Transportmodelle sind in den Dateien „geospheremodel\_2D.h“ und „geospheremodel\_2D.cpp“ implementiert. Das Modelattribut „ModelType“ ist bereits durch die Modellauswahl im ersten Schritt („using Rockflow solver“ oder „using OGS solver“) festgelegt. „InputFileName“ ist ein weiteres Attribut und erhält automatisch den Wert „systeminput“. Auf der Benutzeroberfläche werden direkt nach der Modellauswahl im gleichen Fenster die Anzahl der Geoformationen und der betrachteten Radionuklide, sowie der Adsorptionsisotherme definiert. Diese Werte werden zum Aufruf folgender Methoden verwendet: „*geosphereModelSet()*“ für das eindimensionale Modell, „*geosphereModel2D\_Set()*“ für das zweidimensionale Modell. Die zwei Methoden dienen gemeinsam der Erstellung der Transportmodelle mit zwei Fremdcodes, da der strukturelle Objektaufbau der RockFlow- und OGS-Transportmodelle identisch ist. Ebenfalls bei der anfänglichen Modellerstellung werden die Objekte „geoformation“ und „geologic anomaly“ (existiert nur im zweidimensionalen Modell) von der Methode „*geosphereModel2D\_Set()*“ erzeugt. Die für diese Objekte nötigen Parameter werden anschließend im Objekt „Parameter“ und „ParameterDeterministic“ angegeben.

Die Parameter, die in den eindimensionalen und zweidimensionalen Transportmodellen als probabilistische Parameter behandelt werden, sind in der folgenden Tabelle eingetragen:

Tabelle 5-1 Probabilistische Parameter für die Simulation eines Transportmodells

Parametername	Einheit	Beschreibung
Porosity	-	Die Porosität einer geologischen Formation im Modell (geologische Information)
Dispersion length	$m$	Die Dispersionslänge einer geologischen Formation im Modell (geologische Information)
Storativity	-	Die Speicherfähigkeit einer geologischen Formation im Modell (geologische Information)
Permeability	$m^2$	Die Permeabilität einer geologischen Formation im Modell (geologische Information)
Rock density	$kg/m^3$	Die Dichte des Gesteins einer geologischen Formation im Modell (geologische Information)
Formation length	$m$	Die Länge einer geologischen Formation im Modell (geologische Information, nur im 1-D Modell probabilistisch)
Diffusion coefficient	$m^2/s$	Der Diffusionskoeffizient

Parametername	Einheit	Beschreibung
Release concentration	$kg/m^3$	Die Freisetzungsrates eines Nuklids am Freisetzungspunkt
Sorptionskoeffizienten ( $k_1, k_2, k_3, k_4$ )	$kg/kg$	Die Sorptionskoeffizienten je nach Sorptionsisotherme $k_1$ bis $k_4$
x-coordinate of the nuclide point source	$m$	X-Koordinate der Nuklidfreisetzungsquelle im Modell
y-coordinate of the nuclide point source	$m$	Y-Koordinate der Nuklidfreisetzungsquelle im Modell (bei zweidimensionalem Modell)
Water flux rate	$m^3/s$	Wasserzufluss an der Nuklidfreisetzungsquelle (hydraulische Information)
Fluid density	$m^3/kg$	Die Dichte des Grundwassers (hydraulische Information)
Fluid viscosity	$Pa \cdot s$	Die Viskosität des Grundwassers (hydraulische Information)
Pressure at the begin of path	$Pa$	Der hydraulische Druck an der linken Seite des Modells (hydraulische Information)
Pressure at the end of path	$Pa$	Der hydraulische Druck an der rechten Seite des Modells (hydraulische Information)

„*geosphereModelSet()*“ und „*geosphereModel2D\_Set()*“ erzeugen die Objekte „parameters“ mit dem Konstruktor aus der Klasse „Parameter“ mit den in Kapitel 5.1.3 vorgestellten Attributen. Um die Geoformationen bei zweidimensionalen Modellen abzubilden, erhält jede einen eigenen Satz Parameter. Diese werden so oft als Attribute des „Model“-Objekts gespeichert, wie dies im „Geoformation“-Objekt festgelegt wurde. Genauso läuft der Prozess für die Definition der Transportspezies, der Aufruf des Konstruktors wird entsprechend der Anzahl der Radionuklide wiederholt. Mit einer „Switch() Statement“ werden die Sorptionskoeffizienten entsprechend der vom Benutzer festgelegten Adsorptionsisotherme definiert. Z.B. mit der Auswahl von „Henry Isotherm“ wird nur  $K_1$  als Parameter konstruiert, bei „Freundlich Isotherm“ wird zusätzlich  $K_2$  als Parameter generiert. Neben den probabilistischen Parametern werden die deterministischen Parameter für die zweidimensionalen Modelle definiert. Die deterministischen Parameter werden in die Klasse „ParameterDeterministic“ eingeordnet, die Attribute dieser Klasse sind in Tabelle 5-2 aufgeführt.

Tabelle 5-2      Deterministische Parameter eines Transportmodells

Parametername	Einheit	Beschreibung
Time step length	<i>Jahr</i>	Die Simulationszeitschrittlänge
Time step number	-	Die Anzahl der Zeitschritte, entscheidet wie lange der Zeitraum der Simulation ist
Formation Width	<i>m</i>	Die Breite (Mächtigkeit) einer geologischen Formation im Modell (im zweidimensionalen Modell)
Formation width element	<i>m</i>	Die Länge der y-Richtung für das Modellelement (im zweidimensionalen Modell)
Formation length	<i>m</i>	Die Länge der x-Richtung des Modells (im zweidimensionalen Modell ist die Länge probabilistisch)
Length Element	<i>m</i>	Die Länge der Modellelemente in x-Richtung
Maximal iteration times pressure	-	Der numerische Kontrollwert, der die maximale Iterationsanzahl der Berechnung (Strömung) steuert
Absolute error, pressure	-	Der numerische Kontrollwert, der bei der Iteration die Größe der Toleranz (Strömung) steuert
Maximal iteration number concentration	-	Der numerische Kontrollwert, der die maximale Iterationsanzahl der Berechnung (Transport) bestimmt
Absolute error, concentration	-	Der numerische Kontrollwert, der bei der Iteration die Größe der Toleranz (Transport) steuert
Decay coefficient	$s^{-1}$	Die Zerfallskonstante jedes betrachteten Nuklids (Anzahl der Parameter von Nuklidanzahl bestimmt)
coordinate of Anomaly	<i>m</i>	Die minimalen und maximalen X- und Y-Koordinatenwerte einer Geoanomalie (Sofern diese im zweidimensionalen Modell existiert)
isAnomaly der Formation i	-	Definiert, dass die Anomalie die gleichen Eigenschaften wie eine im Modell existierende Geoformation aufweist
coordinate of output point	<i>m</i>	Der Punkt, an dem die berechnete Konzentration ausgegeben werden soll

In Tabelle 5-2 sieht man die deterministischen Parameter, die als „ParameterDeterministic“ definiert werden. Die Zeitschritte und Schrittzahl bestimmen die Unterteilung der

Simulationszeit und den betrachteten Zeitraum. Die Modellgeometrie (Modelllänge, Breite usw.) kann beim zweidimensionalen RockFlow und OGS Modell auf RESUS nicht probabilistisch behandelt werden. Das liegt an dem Aufbau der Geometrie innerhalb der RockFlow und OGS-Rechenmodelle. Darin werden alle Punkte und Elemente insgesamt durch Gitterkoordinaten definiert und müssen bei Änderung eines Geometrieparameters neu berechnet werden. Der damit verbundene Rechenaufwand ist in der Praxis nicht zu bewältigen. Das Objekt „Formation Width“ kann im zweidimensionalen Modell entweder als Breite oder als Mächtigkeit der geologischen Formation bezeichnet werden.

Die im Objekt „Model“ gespeicherten Geometrieparameter der Geoformationen wiederholen sich mit der Anzahl der Geoformationen im Modell. Der Parameter der Zerfallskonstanten ist für jedes Isotop charakteristisch und wird deshalb im „Model“-Objekt als deterministischer Parameter festgelegt. Die Positionen der Anomalien werden durch Koordinaten definiert, und sind aus den oben für die Modellgeometrie genannten Gründen nicht probabilistisch simulierbar. Wenn die Geoanomalie die gleichen Eigenschaften wie die existierende Geoformation hat, gibt der Benutzer die Nummer der entsprechenden Geoformation ein, dabei beginnt die Nummerierung mit der untersten Geoformation. Hat die Geoanomalie eigene geologische Eigenschaften, so trägt der Benutzer an dieser Stelle „0“ ein und gibt die Parameter wie vorher bei der Geoformation separat ein.

Wenn die probabilistischen und deterministischen Parameter vollständig festgelegt sind, werden durch den Zufallsgenerator entsprechend der vom Benutzer angegebenen Verteilungen die probabilistischen Parameter erzeugt.

Als nächster Schritt müssen die Eingabedateien für die verwendenden Fremdcodes (RockFlow oder OGS) nach der Modelleinstellung erzeugt werden. Da sich die Formate der Eingabedateien der zwei Fremdcodes unterscheiden, muss das Schreiben für die Eingabedateien durch zwei unterschiedliche Methoden realisiert werden. Der Prozess erfolgt durch die Methode „geosphereModel-InputFile()“ für das RockFlow-Transportmodell und „OGSgeosphereModellInputFile()“ für das OGS-Transportmodell.

Der externe RockFlow-Code benötigt für eine Berechnung zwei Eingabedateien: „dataname.rfd“ und „dataname.rfi“. Die „rfi“-Datei beinhaltet die Geometrie und den räumlichen Gitter-Aufbau des Modells. Jeder Punkt im Modell und die FEM-Elemente des Modells werden in dieser Datei definiert. Die Methode „geosphereModellInputFile()“ berechnet die Länge, Breite, Elementlänge, Elementbreite der Finite-Elemente, und schreibt dann diese Werte mittels „ofstream“ Methoden in die „rfi“-Datei.

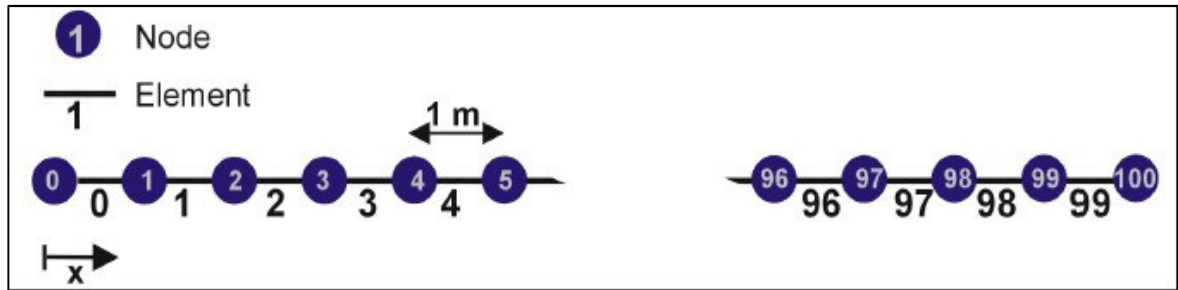


Abb. 5.15a Nummerierungsregel des eindimensionalen Modells in RockFlow und OGS

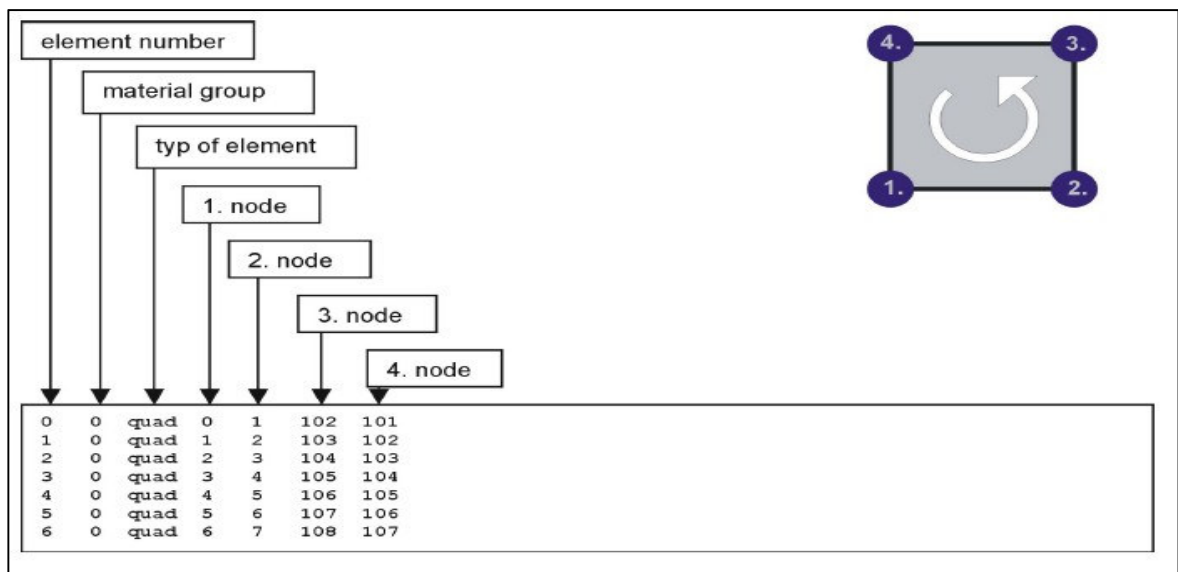


Abb. 5.15b Nummerierung des zweidimensionalen Modells in RockFlow und OGS

Die Geometrie-Eingabedateien von RockFlow und OGS haben das gleiche Format, der Unterschied zwischen den beiden Simulationscodes liegt in der Speicherung aller anderen Modellinformationen. RockFlow arbeitet mit einer einzigen „rfd“-Datei, die alle andere Information außer den Geometriedaten beinhaltet.

```
#SOURCE_VOLUME_FLUID_PHASE
1      ; type
0      ; mode
0      ; curve
0.0 0.0 0.0 ; x,y,z
1.157407e-6; value
```

Abb. 5.16 Ausschnitt der „rfd“-Datei

In Abb. 5.16 legt „type“ mit der Definition des Freisetzungsortes für die Eingabe Freisetzungskordinaten und Konzentrationen nach vordefinierten Typen fest (z.B. Unterscheidung nach Punktquelle, Quelle entlang einer Linie oder einem Flächenprofil); „curve“



entscheidet, ob diese Bedingung zeitabhängig ist; „x,y,z“ sind die Koordinaten, an denen sich die Quelle befindet; „value“ ist die Schadstoffkonzentration, die in dieser Realisierung vorgegeben ist.

In der „rfd“-Datei sind alle Modellinformationen in Modulen angeordnet, die zum Beispiel die Eigenschaften der Gesteine oder der Fluide enthalten. Die Module sind durch sogenannte Marker markiert, welche die Struktur verdeutlichen. So sind zum Beispiel die in Abb. 5.16 gezeigten Modelleinstellungen unter dem Marker „#SOURCE\_VOLUME\_FLUID\_PHASE“ zu finden.

Andere Einstellungen, z.B. die hydraulische oder mechanische Randbedingung sind mit „#BOUNDARY\_CONDITIONS\_PRESSURE“, Stoffparameter mit „#SOIL\_PROPERTIES“ markiert. Die genannte „systeminput.rfd“-Datei für RockFlow wird durch die Methoden „geospheremodel1D\_FileSet()“ und „geospheremodel2D\_FileSet()“ erzeugt, welche die Modelleinstellungen mit den entsprechenden Stichwörtern und ihren Parameterwerten nach immer dem gleichen Schema schreiben.

Statt alle Modelleinstellungen in eine „systeminput.rfd“-Datei mit Stichwörtern zu schreiben, benutzt „OGS“ für unterschiedliche Module separate Dateien. So werden z.B. die Randbedingungen in die „BC“-Datei, die Materialparameter in „systeminput.mfp“-Datei usw. geschrieben. Wenn der Modelltyp „using OGS solver“ gewählt wurde, schreibt die Methode „OGSgeospheremodel1D\_FileSet()“ oder „OGSgeospheremodel2D\_FileSet()“ die Eingabedateien schrittweise mit entsprechenden Parameterwerten und Einstellungen für die OGS-Eingabedateien. Alle Schreibprozesse erfolgt durch das „ofstream“ Objekt.

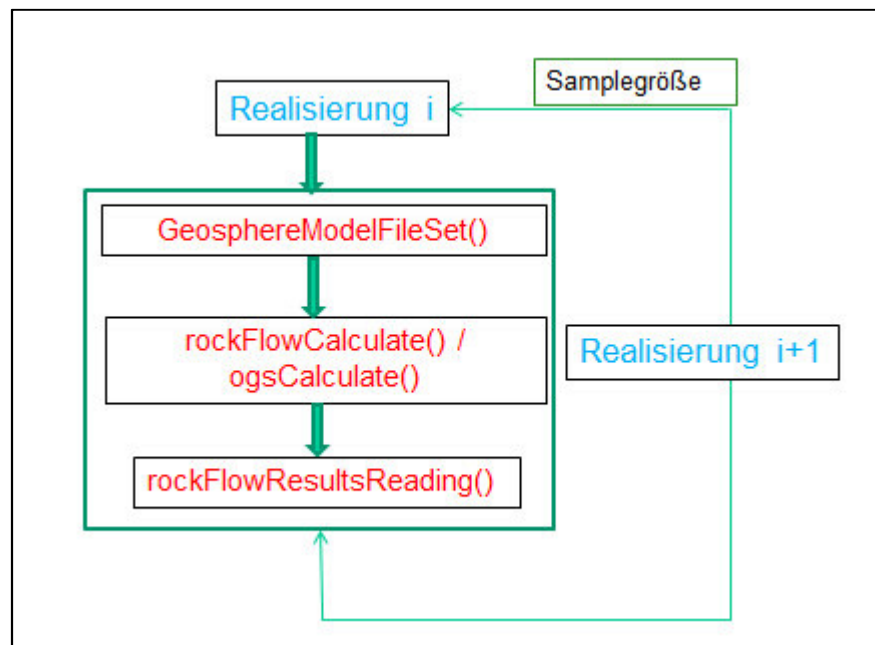


Abb. 5.17 Vorgehensweise der Simulationsphase eines Transportmodells in RESUS

Die in RESUS festgelegte Anzahl der durchzuführenden Simulationsrechnungen mit Hilfe der Fremdcodes (RockFlow oder OGS) werden durch eine „for-Schleife“ realisiert (siehe Abb. 5.17). Eine der vier Methoden: „*geospheremodelFileSet()*“, „*geospheremodel2D\_FileSet()*“, „*OGSgeospheremodel\_FileSet()*“ oder „*geospheremodel2D\_FileSet()*“ wird innerhalb der Schleife zuerst aufgerufen, um die Eingabedateien der entsprechenden Realisierung mittels „ofstream“ zu schreiben. Dann ruft die Methode „*rockFlowCalculate()*“ oder „*ogsCalculate()*“ den Finite-Elemente-Fremdcode auf. Die Methode „*rockFlowResultsReading()*“ findet dann nach der Berechnung die Ausgabedatei und überträgt die Daten mittels „ifstream“ von den entsprechenden Positionen in den Vektor „*calculateResultsValues*“. Die von beiden Fremdcodes berechneten Schadstoffkonzentrationen werden zusammen mit ihren Zeitwerten in der „systemoutput.PLT“-Datei eingetragen. Die Ausgabedateien von RockFlow und OGS haben ein identisches Dateiformat (siehe Abb. 5.18), daher wird die Methode „*rockFlowResultsReading()*“ für beiden Fremdcodes verwendet.

Die drei beschriebenen Methoden werden für eine Realisierung nacheinander aufgerufen. Sobald der Prozess abgeschlossen ist, beginnt die Schleife wieder am Anfang, bis die Anzahl der Rechnungsrealisierungen erreicht ist.

Als in Abb. 5.18 zu sehen enthält die erste Spalte der „.PLT“ Datei die Zeitschrittnummer; die zweite Spalte die Zeitpunkte, für die die Ergebnisse berechnet sind; die dritte Spalte die ausgerechnete Stoffkonzentration am Ausgabepunkt.

```
TITLE = "    RTM 1D (Tracertransport)    "
VARIABLES = "Step ", "Time s" "CONC -"
ZONE T="NODE=216, X=2.160000e+002, Y=0.000000e+000, Z=0.000000e+000, ", F=POINT
0 0.0000000000000000 0.0000000000000000
10 720000000.00000 0.0000000000000000
20 1440000000.0000 0.0000000000000000
30 2160000000.0000 0.0000000000000000
40 2880000000.0000 0.0000000000000000
50 3600000000.0000 0.0000000000000000
60 4320000000.0000 0.0000000000000000
70 5040000000.0000 0.0000000000000000
80 5760000000.0000 0.0000000000000000
90 6480000000.0000 0.0000000000000000
100 7200000000.0000 0.0000000000000000
```

Abb. 5.18 Ausschnitt der Datei „.PLT“

### 5.3.5

#### Die dateibasierte Konstruktionsmethode (file based method)

Wie in den vorangegangenen Kapiteln bereits betont wurde, wurde die Softwareplattform RESUS sowohl für den Einsatz in der Lehre als auch in der Forschung konzipiert. Dieser duale Anspruch wird vor allem bei der Modellerstellung deutlich. Wie schon angesprochen gibt es zwei alternative Erstellungsmethoden: Die GUI-basierte Methode und die dateibasierte Methode. Die GUI-basierte Methode ist für die Verwendung von RESUS in der Lehre wichtig und erlaubt mit überschaubarem Aufwand die Erstellung von Modellen geringer Komplexität. In der Forschung werden jedoch deutlich umfangreichere und flexiblere Simulationsmodelle gebraucht. Sie müssen die komplexen Systeme von Endlagern so genau wie möglich abbilden, um belastbare Aussagen treffen zu können. RESUS ist als Plattform auch für solche Berechnungen geeignet. Sie werden mit Hilfe der dateibasierten Modellerstellungsmethodik angefertigt, deren Funktionsweise im Kapitel 4.5 schon angerissen wurde. In diesem Kapitel soll der Nutzer nach einer Einführung in die schematische Funktionsweise der dateibasierten Methode einen genaueren Einblick in die Struktur der für die Anwendung wesentlichen Dateien bekommen.

Die mit der dateibasierten Methode erstellten integrierten Modelle sind im Gegensatz zu den GUI-basierten erstellten Prozessmodellen in der Lage, mehrere Modellkomponenten zu verknüpfen und bieten auf diese Weise eine größere Flexibilität bei der Modellierung. In RESUS werden die integrierten Modelle mit Submodell-Ketten realisiert, wobei ein Submodell eine Komponente der Endlagersysteme abbildet. Die Submodelle werden von RESUS zu einer Submodellkette mit Parameterbindung aufgebaut. Ein Fremdcode-Algorithmus berechnet darin je eine einzelne Komponente (ein Submodell) des zusammengesetzten Systems (die Submodellkette). Wie in Kapitel 4.5 beschrieben, sind abhängig von den verwendeten Fremdcodes beliebig komplexe Modelle möglich.

Uns interessiert für den Modellaufbau einerseits die in Kapitel 4.5 bereits erwähnte Beschreibungsdatei des Modells, welche die Grundinformationen zum Modellaufbau enthält. Wesentlich sind außerdem zwei weitere Voraussetzungen für die Anwendung der dateibasierten Modellerstellungsmethode zu sehen: Erstens, die Implementierung des benötigten Fremdcodes mit den sogenannten Kommunikationsmethoden in RESUS. Dies ist die Aufgabe der Entwickler. Die Kommunikationsmethoden von RESUS dienen fremdcode-spezifisch als Interface zwischen den Fremdcodes und RESUS. Die zweite Voraussetzung für die Einbindung eines Fremdcodes ist die Erstellung einer fremdcode-spezifischen Template-Eingabedatei durch den Benutzer. Zusammen mit der Beschreibungsdatei enthält sie die wesentlichen Informationen für die Modellgestaltung. Beide Dateien haben feste Formatvorgaben, die bei der Beschreibungsdatei von dem „ModelType“, bei der Template-Eingabedatei durch den Fremdcode festgelegt sind. In der Modellrechnung greift RESUS, wie bereits in Kapitel 4.5.2 angedeutet, für jede Fremdcode-Rechnung auf die Template-Eingabedatei zurück. RESUS wandelt dabei die Template-Eingabedatei in eine fremdcode-spezifische Eingabedatei um, die an den Fremdcode übergeben wird. Sie enthält alle Parameterwerte für die Berechnung, mit denen der Fremdcode

im Anschluss den Rechendurchlauf ausführt. Bei jeder neuen Berechnung tauscht RESUS in diesem Prozess automatisch die probabilistischen Parameterwerte der Eingabedateien aus. Welche Parameter probabilistisch behandelt werden, legt wiederum die Template-Eingabedatei genau fest. Wenn ein einzelner Berechnungsdurchlauf für eine Modellkomponente durch einen Fremdcode durchgeführt wurde, liest RESUS zudem die Ergebnisse aus den Fremdcode-Ausgabedateien aus. Sie werden wie oben beschrieben in einem Ergebnisvektor gesammelt und stehen dann wie schon gezeigt für weitere Analysen auf der GUI bereit.

In diesem Abschnitt soll gezeigt werden, wie die für die Anwendung der dateibasierten Methode wesentlichen Dateien, also die Beschreibungsdatei, die Template-Eingabedatei und die Eingabedatei grundsätzlich aufgebaut sind. Die Kommunikationsmethoden für die dateibasierte Modellerstellungsmethode befinden sich in den Dateien „filtermuster.h“ und „filtermuster.cpp“.

Zunächst betrachten wir die Beschreibungsdatei. In ihr legt der Nutzer den allgemeinen Modellaufbau fest, sie ist folglich von Modell zu Modell verschieden. Als Beispiel für die blockweise Anordnung dient Abb. 5.19.

```
The model name is:
RESUS Example
The model type is:
using Rockflow entire input data with submodels
The model description:
This is an Example of RESUS for a pseudus granite diposal
The number of submodels of this model calculation:
3

The sub-model name is:
fracture-Matrix transport
The model type is:
using RockFlow solver
The model description:
host rock transport model for granite disposal
The name of the input file of the submodel 1:
tangV1.rfd

The sub-model name is:
geosphere Transport 2D
The model type is:
using RockFlow solver
The model description:
Overburdens transport model
The name of the input file of the submodel 2:
asm2d.rfd

The sub-model name is:
Biosphere model
The model type is:
LevelE Biosphere
The model description:
Biosphere model of Level-E
The name of the input file of the submodel 3:
Biosphere.txt
```

Abb. 5.19 Beschreibungdatei eines Dateibasierten Modells

Wir wollen noch einmal den Aufbau der Beschreibungsdatei mit den Grundinformationen des Modells aus Kapitel 4.5 wiederholen, um danach auf ihre Verarbeitung durch RESUS einzugehen. Die vom Nutzer zu erstellende Beschreibungsdatei muss in Textblöcke gegliedert

werden, deren Reihenfolge die Berechnungsreihenfolge und damit die Modellstruktur bestimmt. Der ersten Block enthält stets den Modellnamen, Modelltyp und die Modellbeschreibung des Hauptmodells, sowie die Anzahl der verwendeten Submodelle. Die weiteren Blöcke betreffen jeweils ein Submodell und sind immer gleich aufgebaut. Sie bestehen aus dem Submodellnamen, dem Submodelltyp, der Submodellbeschreibung, sowie dem Namen der Template-Eingabedatei (siehe Abb. 5.19). Die so vorbereitete Beschreibungsdatei wird von RESUS bei der Modellerstellung zuerst verarbeitet. Wir umreißen kurz die verwendeten Methoden. Zunächst liest die Methode „*chainModelBuilding()*“ die Beschreibungsdatei. Unter Zuhilfenahme des „*ofstream*“ und „*ifstream*“ Objekts stellt die Methode „*chainModelBuilding()*“ das Objekt „*Model*“ und dessen „*SubModel*“-Objekte her. In diese Objekte wird die Grundinformation aus der Beschreibungsdatei eingetragen. An dieser Stelle zeigt sich die Fehlerempfindlichkeit der Beschreibungsdatei. Stimmt die Reihenfolge oder Formatierung nicht, so kann die Information nicht korrekt eingelesen werden und die Modellerstellung wird mit einer Fehlermeldung von RESUS abgebrochen. Mit den Namen der Template-Eingabedateien, die in den Submodellen hinterlegt sind, sucht RESUS im nächsten Schritt die zugehörigen Template-Eingabedateien.

Noch einmal machen wir an dieser Stelle einen Einschnitt, um den Aufbau der Template-Eingabedateien zu vertiefen, der in Kapitel 4.5.2 bereits angeklungen ist. Ihr Aufbau ist abhängig von dem verwendeten Fremdcode und variiert von Code zu Code erheblich. Um die Template-Eingabedatei erstellen zu können, greift der Benutzer daher auf die Eingabedatei des betreffenden Fremdcodes zurück, die ihm vorliegen muss. Hier ist für den Benutzer wesentlich, an welcher Stelle dieser Textdatei welche Parameter an den Fremdcode übergeben werden und an welcher Stelle die Ausgabeparameter und Ausgabeorte der Berechnung definiert sind. Diese Kenntnis über die Struktur der angewendeten Fremdcodes ist für den Nutzer vorausgesetzt. An dieser Stelle liegt die eigentliche Hürde bei der Verwendung der dateibasierten Methode. Ist dies aber verstanden, so verläuft die Erstellung der Template-Eingabedatei für alle Fremdcodes gleich. Der Benutzer kopiert zunächst die gesamte Eingabedatei in eine neue Textdatei mit beliebiger Bezeichnung, die zur zukünftigen Template-Eingabedatei wird. Im kopierten Text sind zwei Arten von Eintragungen vorzunehmen. Diese legen sowohl die Berechnung des Fremdcodes als auch darüber hinausgehende Einstellungen der Modellierung fest:

Erstens: Alle Parameterwerte, die in den Rechendurchläufen des Fremdcodes probabilistisch behandelt werden sollen, müssen in der Datei durch den Marker „SIMULATE“ gekennzeichnet werden. RESUS generiert dann an dieser Stelle für jeden Rechendurchlauf selbst die Werte entsprechend der vorgegebenen Verteilung, wie oben beschrieben. Auf diesen Marker erwartet RESUS in der nächsten Zeile die Angaben: deterministischer Wert des Parameters, Parametername, Parametereinheit, statistische Verteilung des Parameters mit Verteilungsmaßen.

Zweitens: Anstelle der Ausgabeparameter mit den Ausgabeorten sind die Marker „RESULT“ zu setzen. Dies funktioniert unter der Voraussetzung, dass jeder Ausgabewert in einer einzelnen Datei gespeichert wird. In den zwei Folgezeilen des „RESULT“-Markers erwartet RESUS die zusätzlichen Angaben (in dieser Reihenfolge): Name der Ausgabedatei, Pfad der Ausgabedatei, Name des Ausgabeparameters, Einheit des Ausgabeparameters, die

Beschreibung des Parameters „ifCouple“ -Wert. Angabe „ifCouple“ (1 steht für „true“ und 0 steht für „false“) entscheidet an dieser Stelle, ob der jeweilige Ausgabeparameter als Kopplungsparameter für das nächste Submodell genutzt werden soll.

```
#SOIL_PROPERTIES
SIMULATE
0.35 ; porosity_matrix - uniform 0.2 0.5

#OUTPUT_EX
;RESULT
tangv1.plt ; path name
CONC ; kg/m3 Concentration nuclide1 in model 1 1
```

Abb. 5.20 Ausschnitt aus einem Beispiel einer Template-Eingabedatei

In diesem Beispiel in Abb. 5.20 ist der Parameter „porosity\_matrix“ probabilistisch und wird durch „SIMULATE“ markiert. Außerdem ist der Ausgabeparameter „CONC“ mit „RESULT“ markiert. Für beide Parameter sind die oben genannten Attribute angegeben. Im nächsten Schritt der dateibasierten Modellerstellung liest RESUS die Template-Eingabedatei Zeile für Zeile mit der Methode „*chainModelBuildingFile()*“ durch. Trifft RESUS darin auf den Marker „SIMULATE“, so wird der Konstruktor der Klasse „Parameter“ aufgerufen, um ein neues Objekt für den probabilistischen Parameter mit den an dieser Stelle hinterlegten Informationen zu erzeugen. Trifft RESUS auf den Marker „RESULT“, so ruft die Methode „*chainModelBuildingFile()*“ den Konstruktor für ein neues „Resultparameter“-Objekt auf, welches so dann mit den angegebenen Attribut-Objekten erstellt und mit den nebenstehenden Werten gefüllt wird.

Auf diese Weise werden von RESUS alle in der Beschreibungsdatei aufgeführten Template-Eingabedateien nacheinander abgearbeitet, bis die Objektgruppen der probabilistischen Parameter und der Ergebnisparameter vollständig sind. Die anschließende Zufallsgenerierung der probabilistischen Parameter erfolgt wieder auf der Nutzeroberfläche durch den gleichen Prozess wie bei Modellen, die über die GUI erstellt wurden.

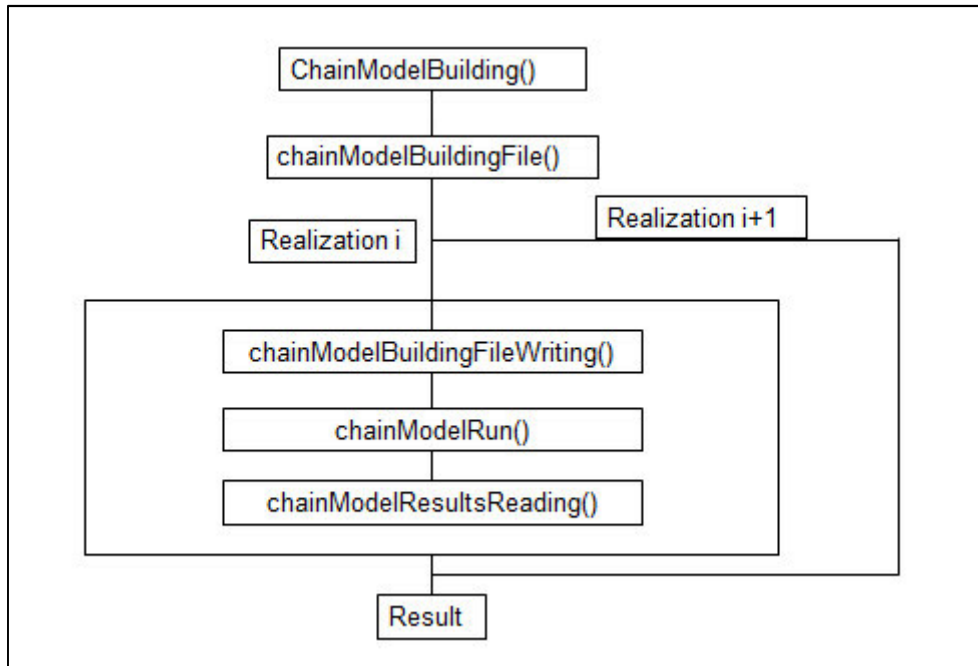


Abb. 5.21 Darstellung des Vorgehens bei der Herstellung eines dateibasierten Modells

Nach der Zufallsgenerierung der probabilistischen Parameterwerte ist das Pre-Processing abgeschlossen. In der folgenden Simulationsphase innerhalb des dateibasierten Modells wird von RESUS zunächst die Transformation der Template-Eingabedatei in die Eingabedatei vorgenommen. Um alle Submodelle einer Modellkette durchzugehen und zum anderen die vorgegebene Anzahl an Realisierungen auszuführen, wird eine zweifach geschachtelte for-Schleife bei der Simulationsphase verwendet. In der Inneren Schleife werden die Methoden „*chainModelBuildingFileWriting()*“, „*chainModelRun()*“ und „*chainModelResultsReading()*“ für jedes einzelne Submodell, das von der äußeren Schleife mit Anzahl der Rechenrealisierungen abgearbeitet wird, aufgerufen.

Die Methode „*chainModelBuildingFileWriting(Model \*model, int submodellIndex, int samplingIndex)*“ wird für die gesamte Simulationsphase benötigt. Die darin verwendete Variable „*samplingIndex*“ gibt den Index der aktuellen Rechenrealisierung an. Die Methode „*chainModelBuildingFileWriting()*“ erzeugt „I/O stream“ zum Lesen und Schreiben der Dateien. Der „*ofstream*“ (Lesestream) heißt „*buffer1*“, der „*ifstream*“ (Schreibstream) heißt „*buffer2*“. Der „*buffer1*“ öffnet im Transformationsvorgang die Template-Eingabedateien und liest diese zeilenweise ein, während der „*buffer2*“ aus dieser Information zeitgleich eine andere Datei namens „*systeminput*“ erstellt. Solange der Lesestream nicht auf den Marker „*SIMULATE*“ trifft, wird der Text der Template-Eingabedatei unverändert in die „*systeminput*“ Datei kopiert. Bei „*SIMULATE*“ aber schreibt „*buffer2*“ anstelle des Markers „*SIMULATE*“ den passenden Samplewert. Dieser wird aus dem Vektor „*samplingValue*“ für diesen Rechendurchlauf entnommen. Der passende Fremdcode für einzelnes Submodell wird in der Methode „*chainModelRun()*“ aufgerufen, z.B. „*runLevele()*“ für ein Level-E Submodell, „*rockFlowCalculate()*“ für ein RockFlow-Submodell usw. „*chainModelResultsReading()*“ dient dem Einlesen der Rechenergebnisse in den „*ResultsParameter*“.

Sofern das Simulationsmodell aus einer Modellkette aufgebaut ist, müssen die Rechenergebnisse jedes darin enthaltenen Submodells an das nachfolgende Submodell weitergereicht werden. Die Weiterleitung der Ergebnisse wird in der Template-Eingabedatei jedes Submodells für jeden Ausgabeparameter separat durch die „ifCouple“-Einstellung festgelegt. Der „ifCouple“-Wert „true“ bedeutet, dass das im Vektor „calculateResultsValues“ gespeicherte Rechenergebnis des betreffenden Parameters (normalerweise eindimensionale zeitabhängige Daten) von RESUS als Eingabedaten in die Eingabedateien für das nächste Submodell eingetragen wird. Die Eintragung übernimmt in RESUS die Methode „chainModelBuildingFileWriting(Model \*model)“. Sie muss von dem RESUS-Entwickler an jeden verwendeten Fremdcode angepasst werden, da die Eingabedateien stets anders formatiert sind. Bei den hier behandelten Fremdcodes RockFlow und OGS beispielsweise werden die eingetragenen zeitabhängigen Randbedingungen durch den Marker „CURVES“ markiert, wie in Abb. 5.22 zu sehen ist. Der beispielhafte Ausschnitt aus der RockFlow-Eingabedatei in Abb. 5.22 enthält die vom vorherigen Submodell übergebenen Randbedingungen der Schadstoffkonzentration. Diese haben einen zeitlichen Kurvenverlauf, der durch einzelne Werte unter „CURVES“ beschrieben wird. Der Kurvenverlauf ersetzt die sonst konstanten Randbedingungen der Schadstoffkonzentration und wird hierzu in der unterstrichenen Zeile referenziert. Die Werte (value1 bis value4) unter dem „CURVES“ werden mit dem Wert(value0) multipliziert und ergeben sich als einzusetzende zeitabhängige Konzentrationsrandbedingung.

```
#BOUNDARY_CONDITIONS_CONCENTRATION
1          ; type
0          ; mode
1          ; curve
0.0 0.0 0.0 ; x1,y1,z1
0.001      ; radius
1.0        ; value0

#CURVES
0.0          1.0 ; time1, value1
43200.0 1.0 ; time2, value2
43201.0      0.0 ; time3, value3
518400.0     0.0 ; time4, value4
```

Abb. 5.22 Eingabedatei für RockFlow mit zwei Randbedingungen: oben „BOUNDARY\_CONDITIONS\_CONCENTRATION“ aus der Template-Eingabedatei und unten „CURVES“ aus dem vorherigen Submodell



### 5.3.6 Datenaufbereitung

Für die statistische Analyse im Post-Processing müssen die in der Simulation gewonnenen Ergebnisse aufbereitet werden. Die dafür nötigen Aufbereitungsmethoden werden in diesem Kapitel vorgestellt. Sie befinden sich in den Dateien „datacondition.h“ und „datacondition.cpp“.

Zunächst ist zu klären, in welcher Art die Ergebnisdaten vorliegen. Wie in den vorangegangenen Kapiteln schon angedeutet, werden alle in der Simulationsphase errechneten Ergebnisdaten im Vektor „calculateResultsValues“ des Objekts „resultsParameter“ gespeichert. Der Vektor „calculateResultsValues“ hat das Format eines zweidimensionalen Vektors mit einer Zeitdimension und einer Dimension für die Realisierungen. Erstere enthält die Zeitschritte der Wertemessung, letztere die zugehörigen Rechendurchläufe.

Im Weiteren wollen wir uns der Berechnung der statistischen Quantile widmen, die in der Analyse von großer Bedeutung sind. Um die statistischen Quantile aus den Ergebnisdaten zu erhalten, müssen in einem ersten Schritt die Ergebnisdaten der Größe nach sortiert werden. Dies geschieht am Anfang des Post-Processings und wird hier zusammen mit den folgenden Verarbeitungsmethoden beschrieben. Zunächst wird ein zweidimensionaler Vektor namens „calculateResultsValuesForStatistic“ von RESUS bei dem Aufruf der Methode „dataCondition(Model \*model)“ erstellt. Die Methode kopiert nun die Ergebnisdaten aus „calculateResultsValues“. In „calculateResultsValuesForStatistic“ werden nun die Daten der Größe an jedem Zeitschritt nachsortiert. Die statistischen Quantile können daraus leicht über die Position in der Datenreihe berechnet werden. Dies geschieht für jedes Quantil mit einer eigenen Methode: Das minimale Quantil wird durch die Methode minimalCalculation() in einem eindimensionalen Vektor namens „minimalValue“ des resultsParameter-Objekts gespeichert. Das 100%-Quantil wird nach gleichem Muster von der Methode maximalCalculation() im Vektor „maximalValue“ hinterlegt. Die 25%- und 75%-Quantile werden mit den Methoden „firstquartileCalculation (Model \*model, int parameterIndex)“ und „thirdquartileCalculation (Model \*model, int parameterIndex)“ errechnet. Diese speichern zunächst die Position des 25%- und 75%-Quantils anhand der Gleichungen „samplingSize“/4 und „samplingSize“×3/4 und sortieren dann die jeweiligen Ergebniswerte dieser Positionen bei allen Zeitschritten in die eindimensionalen Vektoren „firstQuartile“ und „thirdQuartile“ ein. Zur Rundung für die Berechnung der 25%- und 75%-Quantile wird die C++ Methode „int(double)“ verwendet, die eine Zahl ihre Nachkommastellen abrunden lässt.

Es bleibt schließlich noch die Berechnung des Medians (50%-Quantil), der definitionsgemäß von der Anzahl der Rechenergebnisse (bei einem beliebigem Zeitschritt) abhängt. Ist die Anzahl ungerade, so ist der Median der Wert in der Mitte. Ist sie gerade, wird der Median als arithmetisches Mittel der beiden mittleren Zahlen definiert. Diese Berechnung nimmt die Methode medianCalculation(Model \*model, int parameterIndex) vor, die zunächst die Anzahl der Rechenergebnisse (SamplingSize) prüft. Der erhaltene Median von allen Zeitpunkten wird in den eindimensionalen Vektor „median“ des „ResultsParameter“-Objekts gespeichert.

Neben den Quantilen interessieren uns noch die Mittelwerte und die Standardabweichungen für jeden Zeitschritt. Erstere werden von RESUS durch die Methode *meanCalculation(model, parameterIndex)* entsprechend der Gleichung 5-12 berechnet:

$$mean[timeStep] = \frac{1}{n} \sum_{i=0}^{n-1} calculateResultsValuesForStatistic[i][timeStep] \quad (5-12)$$

Im Anschluss werden die Standardabweichungen mit der Methode „*stdvCalculation (Model \*model, int parameterIndex)*“ kalkuliert, entsprechend der folgenden Gleichung 5-13:

$$stdv[timeStep] = \sqrt{\sum_{i=0}^{n-1} \frac{1}{n-1} (calculateResultsValuesForStatistic[i][timeStep] - mean[timeStep])^2} \quad (5-13)$$

wobei ist *mean[timeStep]* der von Methode *meanCalculation()* berechnete Mittelwert zu einer bestimmten Zeit. *n* steht für die Größe der Samples, die als „*Samplingsize*“ in RESUS bezeichnet ist. Die allgemeinen Standardabweichungen werden nach Gleichung 5-13 bestimmt und in den eindimensionalen Vektoren „*Sdev*“ des „*ResultsParameter*“-Objekts geschrieben.

Nach dieser kurzen Auseinandersetzung mit den statistischen Kennwerten interessiert uns im Folgenden die Aufbereitung der Ergebniswerte für die nicht zeitabhängige statistische Analyse. Damit ist in RESUS speziell die Sensitivitätsanalyse gemeint, die bereits in Kapitel 4 vorgestellt wurde. RESUS verfügt für ihre Durchführung verschiedene Ergebnisaufbereitungsmethoden, die hier „*Quantity of interest*“ genannt werden. Die Berechnung der „*Quantity of interest*“ wurde in Kapitel 5.1.4 bereits ausführlich besprochen. Wir wollen hier auf die dafür nötigen programmierten Methoden genauer eingehen.

Wie in Kapitel 5.1.4 und Kapitel 4.1 bereits erläutert, sind in RESUS zwei „*Quantity of Interest*“ implementiert, diese heißen „*Max per realization*“ und „*Integral*“. Die von ihnen berechneten Werte werden als Vektoren gespeichert, die Anzahl der Werte entspricht bei jedem der Gesamtzahl der Rechenrealisierungen des Modells (*Samplingsize*). Wie schon in Kapitel 5.1.4 angedeutet, wird jeder „*Quantity of interest*“ durch eine eigene Methode berechnet: „*Max per realization*“ durch die Methode *maxProRealisierung (Model \*model, int parameterIndex)*, „*Integral*“ durch die Methode *integralCalculation (Model \*model, int parameterIndex)*.

Wir betrachten zunächst die Berechnung des „*Max per realization*“. Für diese wird der oben bereits genannte Ergebnisvektor „*calculateResultsValuesForStatistic*“ auf die bekannte Weise erstellt. Mit den darin befindlichen Ergebniswerten wird der Vektor „*maxPerRealization*“ nach Gleichung 5-14 gefüllt. Er wird als Teil des „*ResultsParameter*“-Objekts gespeichert.

$$maxPerRealization[i] = \text{Max}_{t=0}^{timeSteps-1} calculateResultsValuesForStatistic[i][t] \quad (5-14)$$

wobei *i* in der Gleichung den Sampling-Index angibt.

Die Gleichung 5-14 gibt an, dass im „*calculateResultsValuesForStatistic*“-Vektor aus den Zeitreihen jeweils der maximale Wert herausgesucht wird, siehe auch Abb. 5.23.

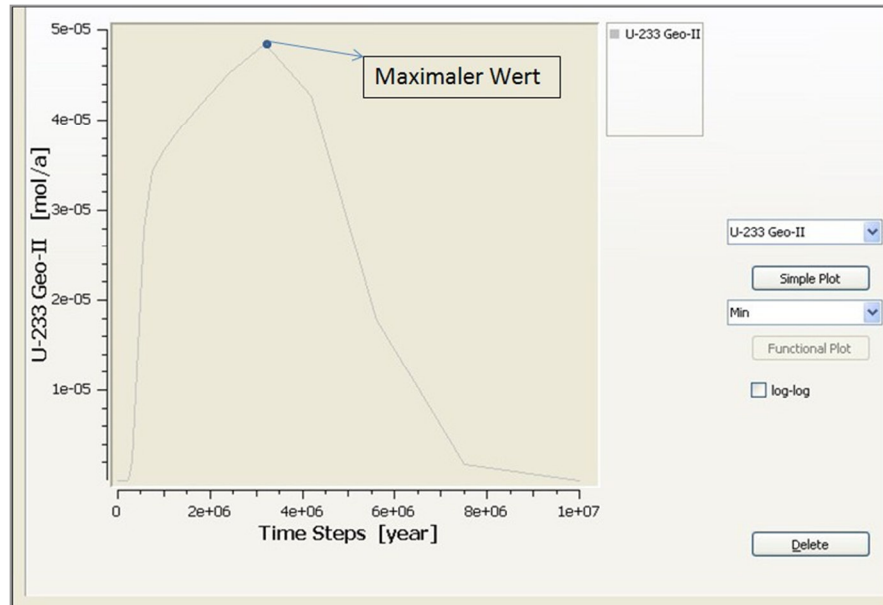


Abb. 5.23 Ein Beispiel eines Maximalen Wertes einer Realisierung

Weiter widmen wir uns dem „Integral“. Bei seiner Verwendung integriert die Methode *integralCalculation* (*Model \*model, int parameterIndex*) die Ergebnisdaten nach Gleichung 5-15. Diese Werte dann im Vektor „Integral“ des Objekts „ResultsParameter“ gespeichert.

$$\text{integral}[\text{sampleIndex}] = \sum_{t=0}^{\text{timeSteps}-1} (\text{Wert}[\text{sampleIndex}][t] + \text{Wert}[\text{sampleIndex}][t + 1]) * \Delta t / 2$$

(5-15)

Für die Berechnung der statistischen Kennwerte für die eindimensionale „Quantity of interest“ wurden die Methoden in der Datei „datacondition.cpp“ implementiert, die für die Datenaufbereitung gebraucht werden. Die Methoden nehmen den Vektor „toCondition“ auf, sortieren die Reihenfolge des Vektors nach der Datengröße und liefern die statistischen Kennwerte als Rückgabewerte aus.

*double findTheHighest(Vector<double> toCondition)* für den maximalen Wert

*double findTheLowest(Vector<double> toCondition)* für den minimalen Wert

*double findTheMean(Vector<double> toCondition)* für den Mittelwert

*double findtheFisrtQuartile(Vector <double> toCondition)* für das 25%-Quantil

*double findTheMedian (Vector <double> toCondition)* für den Median

*double findTheThirdQuartile (Vector <double> toCondition)* für das 75%-Quantil

### 5.3.7 Die grafische Darstellung

Das Post-Processing, also die Auswertung der Rechenergebnisse, wird in RESUS unabhängig von der Art der Modellerstellung mithilfe der grafischen Oberfläche durchgeführt. Dabei werden verschiedene Methoden für die Darstellung von Zeitreihen, Streudiagrammen und Histogrammen benutzt, die alle aus einer externen C++ Bibliothek namens Qwt [61] stammen. Diese Bibliothek gehört zu der Entwicklungsumgebung „Qt“ und enthält eine Fülle von Darstellungsklassen. RESUS speichert die daraus benötigten Methoden in den Dateien „mainWindow.h“ und „mainWindow.cpp“. Auch die grafische Darstellung der RESUS-GUI im Allgemeinen wird durch diese Dateien bestimmt.

Für die Darstellung der Zeitverläufe der Parameter und der Zeitverläufe der statistischen Quantile wird die Klasse „QwtPlotCurve“ genutzt. Jeder einzelne Verlauf wird zunächst als Objekt dieser Klasse erzeugt. Dieses Objekt muss dann mit den Daten des Verlaufs gefüllt werden. Dazu wird die Qwt-Methode *setSamples()* [61] von RESUS aufgerufen. Der Zeitschritt-Vektor und der Ergebnis-Vektor werden als Daten für die x-Achse und die y-Achse eingefügt. Dann wird das erzeugte Objekt des Kurvenverlaufs an die Benutzeroberfläche von RESUS weitergegeben.

Die Bezeichnungen der x- und y-Achse in den grafischen Darstellungen werden von RESUS ohne zusätzliche Benutzereingaben sofort mit Parameternamen und Einheiten angezeigt. Die dafür verwendete Qwt-Methode heißt *setAxisTitle(String)*. Sie gibt die Namen und Einheiten aus dem „ResultsParameter“-Objekt an die Darstellungsmethoden weiter.

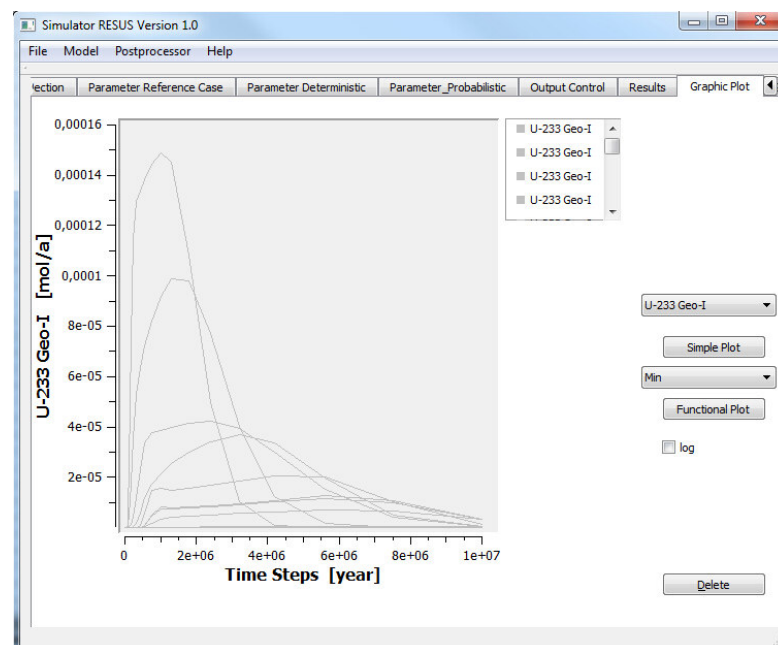


Abb. 5.24 Darstellung eines Ergebnisparameters mit 30 Realisierungen aus Level-E

Jeder Verlauf in Abb. 5.24 stellt eine Ergebnisreihe der entsprechenden Rechenrealisierung dar und wird für die Darstellung als separates Objekt erzeugt. Für die Darstellung der Zeitreihenverläufe wird die logarithmische Transformation als Option angeboten. Dabei werden lediglich die zeitabhängigen Daten logarithmisch transformiert, die Zeitschritte bleiben unverändert. Da null- und negative Werte nicht logarithmisch transformiert werden können, werden diese beim Streudiagramm auch durch den kleinsten transformierten Wert der Datenreihe ersetzt.

Zur weiteren statistischen Analyse kann RESUS die Verteilung der Ergebnisdaten in einem Histogramm anzeigen. Die dahinter stehenden Rechnungen und Darstellungsmethoden sollen hier genauer erläutert werden.

Zunächst ist für die Histogrammberechnung die Reduktion der Ergebnisdaten nötig, welche als Zeitverläufe für die einzelnen Durchgänge vorliegen. Wie oben erläutert, kann dies wahlweise durch die „Quantity of interest“: „max per realization“ oder „integral“ geschehen. Die so ermittelten Histogrammergebnisse werden dann in einen Vektor gespeichert (entweder in „maxPerRealization“ oder „Integral“). Die Anzahl der Wertklassen wird durch den Benutzer in der GUI bei „number of bins“ festgelegt. Mit dieser Festlegung erfolgt die Berechnung der Klassenbreite aus dem maximalen und minimalen Wert. Dann sind die Histogrammklassen ausreichend bestimmt und jedes Histogrammergebnis wird in eine der Wertklassen einsortiert. RESUS ermittelt daraufhin in einer Schleife, wie häufig jeder Index vorkommt und wir erhalten die damit absolute Häufigkeit für jede Wertklasse. Sie wird im Feld „intervalUnitCount[number of bins]“ gespeichert.

Bei der Erstellung des Histogramms werden die statistischen Quantile (minimaler Wert, 25%-Quantil, Median, 75%-Quantil und maximaler Wert) mit Hilfe der oben aufgeführten Berechnungsmethoden in der Datei „datacondition.cpp“ ermittelt. Sie werden, wie in Kapitel 4 beschrieben, auf der Benutzeroberfläche angezeigt. Überdies besteht wie schon bei den Zeitreihenverläufen auch beim Histogramm die Möglichkeit, die Daten für die Darstellung logarithmisch zu transformieren. Die Wertklassen werden ebenfalls logarithmisch transformiert. Wenn Null oder negative Werte in der Datenreihe erscheinen, macht die logarithmische transformierte Histogrammdarstellung keinen Sinn, Daher wird ein Warnungsfenster bei der GUI aufgerufen und die Transformation wird nicht durchgeführt.

Das Werkzeug für die Sensitivitätsanalyse in RESUS ist die Streudiagramm-darstellung (Scatterplot). Sie wurde auf RESUS ebenfalls mit den Darstellungsmethoden aus der Qwt-Bibliothek [61] umgesetzt. Dieses Darstellungsverfahren greift wieder auf die „Quantity of interest“ zurück. Die auf diese Weise aufbereiteten Ergebnisparameter werden in einem Zwei-Achsen-Koordinatensystem gegen die Werte eines wählbaren probabilistischen Eingangsparameters aufgetragen.

Die Klasse „QwtPlotCurve“ ist für die Darstellung der Werte zuständig. RESUS speichert die Samplewerte der probabilistischen Eingangsparameter und die errechneten „quantity of interest“ der Ausgabeparameter in Form eines Arrays. Das Objekt „QwtPlotCurve“ wird dann an eine Plotter-Methode übergeben, die die Darstellung realisiert. Gleichzeitig werden durch die

Methoden *setTitel()* des Qwt-Werkzeugs die Achsentitel mit Parameternamen und Einheit für die Darstellung festgelegt.

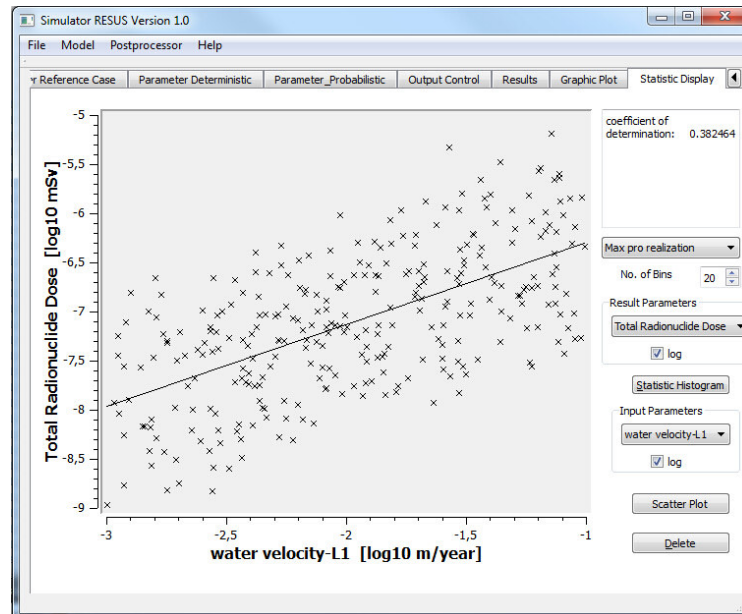


Abb. 5.25 Beispiel eines Streudiagramms in RESUS. Das Beispiel entstammt der Level-E-Modellrechnung

In Abb. 5.25 kann man sehen, dass RESUS sowohl die Regressionslinie als auch das Bestimmtheitsmaß des Streudiagramms berechnet und darstellt. Für die Berechnung dieser zwei statistischen Maße ist die Klasse „Regression“ zuständig.

Die Regressionsgerade und die Regressionsmaße zwischen den gegeneinander aufgetragenen Ein- und Ausgabeparameterdaten werden nach folgenden Gleichungen gerechnet:

Die Trendlinie:

$$y = Ax + B \quad (5-16)$$

$$B = S_{xy}/S_{xx} \quad (5-17)$$

$$A = \bar{y} - B/\bar{x} \quad (5-18)$$

hier ist  $\bar{y}$  der Mittelwert der „Quantity of interest“ des Ausgabeparameters,  $\bar{x}$  ist der Mittelwert der probabilistischen Eingangswerte. A ist die Steigung der Regressionslinie und B der Schnittpunkt der Linie mit der y-Achse.

$$S_{xy} = \sum_{i=1}^{samplingSize} ((x_i - \bar{x})(y_i - \bar{y})) \quad (5-19)$$

Die Berechnung des Bestimmtheitsmaßes erfolgt nach:

$$R^2 = \frac{s_{xy}^2}{s_{xx} \times s_{yy}} \quad (5-20)$$

hierbei ist  $S_{xy}$  die Kovarianz der Ein- und Ausgabeparameter,  $S_{xx}$  ist die Varianz der Eingabeparameterwerte,  $S_{yy}$  ist die Varianz der „quantity of interest“ des Ausgabeparameters.

Die Klassendefinition der „Regression“-Klasse befindet sich in den Dateien „regression.h“ und „regression.cpp“. Die Variablen werden als private Attribute in der Klasse mit *set()* und *get()* Methoden definiert. Die enthaltenen Attribute werden in der Tabelle 7-8 im Anhang aufgelistet.

Bei der Streudiagrammdarstellung greift der Konstruktor der Klasse automatisch den „Quatity of interest“-Vektor und den Samples-Vektor des probabilistischen Parameters auf. Die mathematischen Berechnungen der Variablen werden in *set()*-Methoden durchgeführt. Die Zwischenergebnisse der Rechnungen werden mit *get()*-Methoden zur weiteren Kalkulation aufgerufen. Die berechnete Regressionslinie wird in die Streudiagrammdarstellung gezeichnet und das Bestimmtheitsmaß wird ins Fenster daneben eingetragen.

### 5.3.8 Speichern und Wiederöffnen eines RESUS-Modells

Die Struktur eines in RESUS erstellten Modells kann vor und nach der Berechnung in Dateien gespeichert und wieder in RESUS eingelesen werden. An diesem Speicher- und Leseprozess ist eine Reihe von Methoden zu beteiligen. Sie sind in den Dateien „SaveAndLoad.h“ und „SaveAndLoad.cpp“ implementiert und werden zur Übersicht vollständig aufgezählt:

Die Methode *saveFile()* wird aufgerufen, sobald der Benutzer in der GUI das Speichern des Modells veranlasst. Je nachdem ob vorher schon einmal gespeichert wurde, öffnet oder erzeugt *saveFile()* eine Textdatei und fragt in letzterem Fall nach dem Namen. Die Objekte und Attribute des „model“-Objekts werden zuerst in Dateien mit dem Schreibstream („ofstream“) in ASCII-Text gespeichert. Dieser Text sieht wie in folgendem Beispiel aus:

```

The model name is:
RESUS Example
The model type is:
using Rockflow entire input data with submodels
The model description:
This is an Example of RESUS for a pseudus granite diposal
the parameter number:
10
the time step number:
1001
the time unit:
year
the sampling size:
300
The number of Geoformations:
0
The number of components:
1
The Sorption concept Index:
0
If the decay chain of nuclides should be considered:
0
If the model has been calculated:
1

```

Abb. 5.26 Textbeschreibung der Objekte und Attribute des „model“-Objekts in der zugehörigen Textdatei

In der gleichen Datei werden zusätzlich zu den Grundattributen des „model“-Objekts auch die Attribute des „submodel“-Objekts gespeichert. Sie stehen in der Datei weiter unten, ein Beispiel für den Speichertext des Objekts „SubModel“ ist in Abb. 5.27 zu sehen.

```

The model name is:
matrix diffusion model
The model type is:
using rockflow solver
The model description:
2-D rock flow matrix diffusion
The name of the input file of the submodel 1:
tangV1

```

Abb. 5.27 Textbeschreibung der Objekte und Attribute des Objekts „Submodel“ in der zugehörigen Textdatei

Im nächsten Schritt werden in der gleichen Textdatei die probabilistischen und deterministischen Parameter hinterlegt. Sie folgen wiederum den Zeilen für das Submodel-Objekt. Ein Beispiel für ihre Formatierung ist in Abb. 5.28 gegeben. Gespeichert werden unter anderem Name, Einheit, Beschreibung. Diese Abfolge der Parameterattribute wiederholt sich für jeden Parameter.



```

the parameter named:
permeability
the parameter unit:
m2
the parameter description:
permeability_of_fracture
the probabilistic distribution:
log_uniform
the parameter value:
8e-010
the min value:
1e-010
the max value:
8e-009
the probabilistic MinValue:
1e-010
the probabilistic MaxValue:
8e-009

```

Abb. 5.28 Ausschnitt aus der Speicher-Textdatei des Model-Objekts

Die Zeitschritte werden bei RESUS als probabilistische Parameter in einem „samplingValue“-Vektor gespeichert. Der Inhalt dieses Vektors wird im Speicherprozess an dieser Stelle zuuntermst in die Textdatei eingefügt. Ein Beispiel findet sich in Abb. 5.29. Zu sehen sind die Speicherzeilen der Zeitschritte der Simulation, welche auch die Ausgabe der Ergebnisdaten festlegen.

```

the time steps:
1300  1800  2400  3200  4200  5600  7500  10000  13000  18000
24000 32000 42000 56000 75000 100000 130000 180000 240000
320000 420000 560000 750000 1e+006 1.3e+006 1.8e+006 2.4e+006
3.2e+006 4.2e+006 5.6e+006 7.5e+006 1e+007

```

Abb. 5.29 Ausschnitt aus der Speicher-Textdatei des Model-Objekts

Die Vektoren aus dem „samplingValues“-Objekt enthalten die zufallsgenerierten Werte aller probabilistischen Eingabeparameter (Inputsamples) und werden in einer zweiten Textdatei gespeichert, um eine weitere Verarbeitung der Daten zu vereinfachen. Sie befinden sich dort in einem reinen Datenblock ohne zusätzliche Beschreibungszeilen, was die Wiederherstellung des Modells beschleunigt. Für diese Speicherung wird die Methode „*saveInputSampling()*“ benutzt, die durch den Aufruf der Methode „*saveFile()*“ ausgeführt werden kann. Die Methode „*saveInputSampling()*“ schreibt den genannten Datenblock von den Samples mit mehreren Realisierungen und benennt die neue Datei mit einem kombinierten Namen aus der Bezeichnung der allgemeinen Beschreibungsdatei und dem Anhang „\_Sample“. Beispielsweise würde diese Datei in einem Modell mit der Beschreibungsdatei „RESUS\_Example.dat“ „RESUS\_Example\_Sample.txt“ heißen. Der Datenblock in der Datei hat ein definiertes Format, das in Abb. 5.30 veranschaulicht wird.

The Input Sample:

```
8e-010 0.35 5e-005 5e-005 0.2 1e-012 5 1e+007 0.73 100000
4.30839 0.203556 0.000279313 8.66455 0.0966733 3.10919 8.09004 1e+007 0.73 100000
```

Abb. 5.30 Ausschnitt aus der „\_Samplings“-Datei, die die gespeicherten Inputsamples in einem Datenblock enthält

Die Daten sind in Matrixform angeordnet, die Zeilen enthalten je einen Parameter, die Spalten je einen Sample-Satz eines Rechendurchlaufs.

Die in einem zu speichernden Modell enthaltenen Ergebnisparameter werden mit der Methode „*saveResultsToFile(Model \*model, QString fileName)*“ gesichert. Diese Methode erzeugt dazu eine weitere Datei oder öffnet eine bereits vorhandene, sofern es bereits einen Speichervorgang gab. Der bei der Erzeugung dieser Datei gewählte Dateiname basiert auf dem Namen der im Modell verwendeten Inputdatei und erhält überdies den Zusatz „\_Results“. Heißt die Input-Datei also beispielsweise „RESUS\_Example“, so wird die Speicherdatei der Ergebnisparameter „RESUS\_Example\_Results“ benannt. Die Methode schreibt in die „\_Results“-Datei mit „ofstream“ Objekt zuerst zusammenfassende Grundinformationen wie u.a. den Modelltyp, Namen, Einheiten und die Anzahl der Ergebnisparameter, die einzelnen Parameterbeschreibungen und bestimmte Parameterattribute.

```
The model name is:
RESUS Example
The model type is:
This is an Example of RESUS for a pseudus granite diposal
the model description:
This is an Example of RESUS for a pseudus granite diposal
the Result Parameter number:
4
the sampling size:
300

the parameter named:
CONC
the parameter unit:
kg/m3
the parameter description:
Concentration_of_nuclide_1_at_model_1
If the parameter should be displayed
1

the parameter named:
CONC
the parameter unit:
kg/m3
the parameter description:
Concentration_of_nuclide_1_at_model_2
If the parameter should be displayed
1
```

Abb. 5.31 Beispielausschnitt einer „\_Results“-Textdatei

In Abb. 5.31 zu sehen ist der obere Bereich dieser Datei mit den Grundinformationen des Modells und der einzelnen Parametern. Hier sind für die dargestellten Parameter die Attribute Name, Einheit, Beschreibung, ifDisplay (ob der Parameter dargestellt werden soll) aufgelistet.

Die „\_Results“-Textdatei wird beim Speichern eines RESUS-Modells in jedem Fall angelegt, auch wenn im Modell noch keine Modellierungsrechnung durchgeführt wurde. Für RESUS ist jedoch die Information, ob eine Modellierungsrechnung stattfand oder nicht, relevant für den weiteren Speicherprozess. Sie wird daher in der „\_Results“-Textdatei im Bereich der Modellattribute als Boolesche Variable namens „ifHasBeenCalculated“ hinterlegt. Ist diese Variable „true“, so hat eine Modellierung stattgefunden und RESUS wird die dabei verwendeten Ergebnisparameter zusätzlich noch in eine separate Datei abspeichern, die auch zur weiteren Ergebnisverarbeitung mit externen Programmen (z.B. Excel) dienen kann. Dies geschieht mit der Methode „*saveSampleResults(Model \*model, QString fileName)*“. Die Ergebnisse werden durch sie in eine Datei gespeichert, deren Name zusammengesetzt ist aus dem vom Benutzer gewählten Modellnamen und dem Zusatz „\_SamResults“. Heißt das Modell beispielsweise „RESUS\_Example“, so werden die Ergebniswerte unter dem Namen „RESUS\_Example\_SamResults“ gespeichert. Gab es bereits einen Speichervorgang, so öffnet diese Methode die bestehende Datei, andernfalls erstellt sie sie mit der beschriebenen Nomenklatur neu. Dann werden darin die Ergebnisdaten mittels des „ofstream“ Objekts als Matrix eingetragen. Ein Datenblock enthält sämtliche berechneten Werte eines Ausgabeparameters. Eine Zeile des Blocks entspricht einem Rechendurchlauf, eine Spalte enthält alle Werte eines Zeitschritts. Zwischen den Blöcken der Ausgabeparameter steht jeweils eine leere Zeile. Auf diese Weise werden auch bei Modellen mit mehreren Komponenten und also mehreren Fremdcodes alle in der Berechnung erhaltenen Ergebnisse und Zwischenergebnisse in einer Datei festgehalten.

Realisierung1(Zeit1)	Realisierung1(Zeit2)	Realisierung1(Zeit3)	Realisierung1(Zeit4)
Realisierung2(Zeit1)	Realisierung2(Zeit2)	Realisierung2(Zeit3)	Realisierung2(Zeit4)
Realisierung3(Zeit1)	Realisierung3(Zeit2)	Realisierung3(Zeit3)	Realisierung3(Zeit4)
Realisierung1(Zeit1)	Realisierung1(Zeit2)	Realisierung1(Zeit3)	Realisierung1(Zeit4)
Realisierung2(Zeit1)	Realisierung2(Zeit2)	Realisierung2(Zeit3)	Realisierung2(Zeit4)
Realisierung3(Zeit1)	Realisierung3(Zeit2)	Realisierung3(Zeit3)	Realisierung3(Zeit4)

Abb. 5.32 Beispielausschnitt einer „\_SamResults“-Textdatei

In Abb. 5.32 zu sehen sind deswegen zwei Datenblöcke getrennt durch eine Leerzeile. Sie enthalten jeweils die Ergebniswerte eines Ausgabeparameters. Jeder Block besteht aus drei Zeilen für drei Rechenrealisierungen, die wiederum die Parameterwerte innerhalb eines Rechendurchlaufs mit vier Zeitschritten.

Zuletzt beschreiben wir den Vorgang des Öffnens eines gespeicherten Modells aus den Speicherdateien und gehen auf die dafür in RESUS programmierten Methoden ein. Durch das Öffnen kann ein einmal behandeltes Modell in RESUS wie vor dem Speichervorgang ohne Einschränkungen weiterverarbeitet, nachgerechnet, analysiert und kontrolliert werden. Es werden also alle Rechnungs- und Ergebnisdaten wiederhergestellt.

Der Benutzer veranlasst das Öffnen über die GUI durch den Knopf „Open old model“ und wählt im daraufhin erscheinenden Fenster die Datei mit der Endung „.dat“. Daraufhin erfolgen die restlichen Vorgänge automatisch. Alle danach aufgerufenen Methoden, die das Öffnen des Modells bewirken, sind zusammen mit den Speicher-Methoden in den Dateien „SaveAndLoad.h“ und „SaveAndLoad.cpp“ implementiert.

Nach der Öffnen-Aufforderung wird zuerst die Methode „*openFile(Model \*model, QString fileName)*“ ausgeführt, welche die Speicherdateien sucht. Mittels des „ifstream“ Objekts wird dann aus den vier beschriebenen Speicherdateien (Speichertextdatei des „Model“-Objekts, „\_Samplings“-Datei, „\_Results“-Textdatei, „\_SamResults“-Textdatei) die Objektstruktur des Modells wiederhergestellt. RESUS benutzt für diesen Prozess eine Methode namens „*loadFile(Model \*model, QString fileName)*“. Wurde bei dem betreffenden Modell bereits eine Simulation durchgeführt, so erkennt RESUS dies am Attribut „ifHasBeenCalculated“ in der eingelesenen „\_SamResults“-Datei. RESUS ruft in diesem Fall die Methode „*loadResultsFile(Model \*model, QString fileName)*“ auf. Diese überträgt die Daten der „\_SamResults“-Textdatei daraufhin in den Vektor „calculateResultsValues“. Nach diesem Vorgang ist das Modell wieder vollständig hergestellt. Analyse für die gespeicherten Modelle können ohne Aufwand weiter durchgeführt werden.

Die vorliegende Arbeit befasst sich mit der Einführung einer neuen Software namens RESUS. Die Softwareplattform wurde vom Verfasser dieser Dissertation konzipiert und programmiertechnisch umgesetzt. Es handelt sich bei dieser Software um ein wissenschaftliches Werkzeug zur numerischen Modellierung im Rahmen von Langzeitsicherheitsanalysen für Endlager. Das hinter RESUS stehende Konzept ist für die duale Nutzung in Forschung und Lehre entwickelt. Die praktische Umsetzung dieses Konzepts in Form einer Plattformstruktur wird auf verschiedene Weise dargelegt. Zum einen erfolgt eine sequentielle Darstellung angefangen von der Modellerstellung über die numerische Simulationsrechnung bis hin zur Aufbereitung und Darstellung der Ergebnisse. Zum anderen wird der Rückgriff auf externe Fremdcodes dargelegt, die RESUS zur Ausführungsplattform für komplexe Modellrechnungen werden lassen. Es wird erklärt, wie auf der grafischen Oberfläche von RESUS Nutzerfreundlichkeit und Flexibilität verbunden werden um die sehr unterschiedlichen Ansprüche aus Lehre und Forschung zu erfüllen. Die Anwendungsmöglichkeiten der RESUS-Software sowohl in der Lehre als auch in der Forschung werden mittels Beispielen vorgestellt und sollen dem Leser in Form eines Tutorials Anhaltspunkte für die Benutzung liefern. Diese Dissertation dient auch als Entwicklungsdokumentation für die Weiterentwicklung der Softwareplattform. Zusätzlich zu diesem Anleitungsteil ist in der vorliegenden Arbeit auch die genaue Funktionsweise von RESUS dargestellt.

Die Modellerstellung für RESUS erfolgt durch zwei Methoden: Zum einen der Benutzeroberflächen-basierten (GUI) Methode und zum anderen der dateibasierten Methoden. Es wird in dieser Arbeit aufgezeigt, wie die beiden Methoden in RESUS funktionieren und für welche Problemstellungen die Methoden am besten geeignet sind. So besteht ein erheblicher Unterschied zwischen den genannten Methoden hinsichtlich der Anforderungen an den Benutzer und damit einhergehend der Komplexität und Flexibilität der erstellbaren Modelle. Die GUI-basierte Methode ist für den Lehrbetrieb gedacht und soll auf einfache Weise Lehrbeispiele simulierbar machen. Die dateibasierte Methode hingegen soll zur Erstellung komplexer Modellstrukturen für die Forschung genutzt werden. Dabei sind Modelle mit der GUI-basierten Methode einfacher und schneller herzustellen als mit der dateibasierten Methode. Mit der GUI-basierten Methode sind ausschließlich die in RESUS fest implementierten Modellarten simulierbar. Neben der GUI-basierten Modellerstellungsmethode kann die dateibasierte Methode eine breite Vielfalt von Modellen simulieren. Der Aufbau der Modelle bzw. sogar Modellketten kann flexibler und komplexer erfolgen, deswegen ist diese Methode eher für die professionellen Wissenschaftler bei der Forschungsarbeit vorgesehen.

Die zentralen Arbeitsschritte der dateibasierten Methode sind:

1. Eine Beschreibungsdatei für die Grundinformation über das Modell und dessen Submodelle wird vom Nutzer geschrieben.

2. Die Template-Eingabedateien für den einzusetzenden numerischen Fremdcode werden zuerst erstellt, wobei die probabilistischen Parameter und Ausgabeparameter gekennzeichnet werden sollen.
3. RESUS liest die Beschreibungsdatei. Mit der Information in der Datei findet RESUS die Template-Eingabedateien für jedes Submodell. RESUS liest die Template-Eingabedateien einzeln, erkennt die probabilistischen Parameter und Ergebnisparameter.
4. In RESUS wird das Sample für die probabilistischen Parameter erzeugt. RESUS erstellt die temporären Eingabedateien aus den Template-Eingabedateien für eine Realisierung, in denen die Werte der probabilistischen Parameter an deren Stelle mit entsprechenden zufälligen Werten ersetzt worden sind. Mit den Eingabedateien ruft RESUS dann die Fremdcodes zur Durchführung einer Rechenrealisierung auf.

Die dateibasierte Modellerstellungsmethode ist in dieser Version von RESUS nur einsetzbar, wenn die Eingabedateien der numerischen Fremdcodes textförmig (ASCII) sind. Bei binären Eingabedateien ist die Identifikation der probabilistischen und Eingabeparameter problematisch.

Der gesamte Daten- und Informationsaustausch zwischen RESUS und den Fremdcodes von RESUS selber wird durch Lese- und Schreibstream realisiert. Nach der Berechnung der mit einem der beiden Erstellungsverfahren erzeugten Modelle werden die Ergebnisse mit RESUS dargestellt und analysiert. RESUS bietet die Möglichkeit, die zeitabhängigen Ergebnisdaten zu plotten, logarithmisch zu transformieren, und mit Hilfe von Histogrammen und Streudiagrammen zu analysieren.

Bei der Entwicklung und Anwendung von RESUS sind auch Probleme aufgetreten. Bei mehrmaligem Aufruf der numerischen Fremdcodes könnte es möglich sein, dass numerische Konvergenzprobleme wegen ungünstiger zufälliger Samplekombination auftreten. Wenn Konvergenzprobleme auftreten, könnte die aktuelle Simulation ausfallen, was den gesamten Verlust aller Rechenrealisierungen nach sich ziehen könnte, da die Ergebnisvektoren beim Ausfall keine Daten erhalten können. Das Problem muss bei der nachfolgenden Version von RESUS gelöst werden. Für die beschriebene Version von RESUS gibt es folgende Weiterentwicklungsmöglichkeiten:

- Weiterentwicklung der grafischen Benutzeroberfläche, damit die Arbeit mit RESUS weiter erleichtert werden kann.
- Erweiterung der Flexibilität hinsichtlich der Verwendung von unterschiedlichen numerischen Fremdcodes. Insbesondere betrifft dies Methoden, die der Implementierung der numerischen Fremdcodes mit nicht-ASCII Ein- und Ausgabedateien dienen.
- Implementierung weiteren Methoden der Sampleerzeugung, da mit der vorliegenden Version die Samples nur mit der „Simple Random“ Methode erzeugt werden.
- Erweiterung zur Berechnung weiterer Sensitivitätsmaße, gegenwärtig wird nur das lineare Bestimmtheitsmaß berechnet und dargestellt.
- Schaffung der Möglichkeit, RESUS mit unterschiedlicher Hard- und Softwareumgebung zu betreiben. Diese Version von RESUS funktioniert nur unter MS-Windows.

- Um die Rechenzeit zu reduzieren, muss die parallele Verteilung der Rechenrealisierung auf unterschiedlichen Rechenprozessoren ermöglicht werden. Das ist sinnvoll, wenn die probabilistischen Simulationen auf einem großen Rechenserver statt am PC berechnet werden.
- Eine weitere Anwendungsrichtung der probabilistischen Simulation bei der Langzeitsicherheitsanalyse ist die Arbeit mit inhomogenen geometrischen Feldern, wie sie z.B. in [56] dargestellt ist. Es ist vorgesehen, derartige Analysen auch in RESUS zu ermöglichen.

In diesem Textanhang werden die Tabellen für die Sammlung der Attribute und Methoden dargestellt.

Tabelle 7-1                      Objekte und Methoden der Klasse „Model“.

Objekte	Typ	Beschreibung	Entsprechende Methoden
Name	String	Name vom Modell	setName(String)    getName()
Description	String	Kurze Beschreibung vom Modell als Information	setDescription(String) getDescription()
ModelType	String	welches Fremdprogramm wird verwendet	setModelType(String) getModelType()
timeStepsNumber	Integer	Anzahl der Zeitschritten	setTimeStepsNumber(int) getTimeStepsNumber()
Samplingsize	Integer	Anzahl der probabilistischen Realisierungen	setSamplingSize(int) getSamplingSize()
ifHasBeenCalculated	Boolean	Checken ob das Modell durchgerechnet wurde	setIfHasBeenCalculated(boolean) getIfHasBeenCalculated()
modelSaveFileName	String	Name der RESUS-Inputdatei des Modells	setInputFileName(String) getInputFileName()
ifChain	Boolean	Angabe ob die Nuklide eine Zerfallskette bilden	setIfChain(boolean) getIfChain()
subModelNumber	Integer	Anzahl der Submodelle	setSubmodelNumber(int) getSubmodelNumber()
componentsNumber	Integer	Anzahl der rechnenden Schadstoffkomponenten	setComponentsNumber(int) getComponentsNumber()



Objekte	Typ	Beschreibung	Entsprechende Methoden
adsorptionsIndex	Integer	Entscheidet, welches Sorptionsmodell eingesetzt werden soll	setAdsorptionsIndex(int)  getAdsorptionsIndex()
histogramIntervalNumber	Integer	Anzahl der Intervalle bei der Darstellung von Histogramm	setHistogramIntervalNumber(int)  getHistogramIntervalNumber()
subModel	Vector <SubModel>	subModell als eine Klasse wird im Modell beibehalten	Vererbte Klasse von „Model“, die als Modellkomponente dargestellt werden sollen
Parameter	Vector <Parameter>	Klasse von Eingabeparameter mit probabilistischen Samples	Die Klasse Parameter wird in unteren Kapitel vorgestellt
ParameterDeterministic	Vector<ParameterDeterministic>	Deterministische Parameter	Die Parameter, die nicht probabilistisch gehandelt werden
ResultsParameter	Vector <ResultsParameter>	Klasse von Ergebnisparametern	Vererbte Klasse von „Parameter“, die als Zwischen- oder Endergebnis betrachtet wird
timeSteps	Parameter	Die Zeitschritte für die Simulation	Timesteps, die als Objekt zur Klasse Parameter gehört.
Geoformation	Vector <Geoformation>	Die zu simulierenden Geoformationen, jede als ein Objekt betrachtet	Als eine Klasse genutzt für eine Geologische Schicht
GeoAnomaly	Vector <GeoAnomaly>	inhomogener Bereich in Geoformation	Als eine Klasse genutzt für die „Lücke“ in Geoformation

Tabelle 7-2

Objekte und Methoden der Klasse „SubModel“

Objekte	Typ	Beschreibung	Entsprechende Methoden
Name	String	Name des Sub-Modells	setName(String) getName()
Description	String	Kurze Beschreibung des Sub-Modells	setDescription(String) getDescription()
ModelType	String	Der entsprechende numerische Löser wird dadurch gefunden	setModelType(String) getModelType()
InputFileName	String	Name der Template-Eingabedatei	setInputFileName(String) getInputFileName()
timeStepsNumber	Integer	Anzahl der Zeitschritte bei Simulation	setTimeStepsNumber(int) getTimeStepsNumber()
ifHasBeenCalculated	Boolean	Checken Ob das Submodell durchgerechnet wurde	setIfHasBeenCalculated(boolean) getIfHasBeenCalculated()
parameterNumber	Integer	Anzahl der Eingabeparameter	setParameterNumber(int) getParameterNumber()
resultsParameter-Number	Integer	Anzahl der Ausgabeparameter	setResultsParameterNumber(int) getResultsParameterNumber()
timeSteps	Parameter	Simulationszeitschritte	Timesteps, die als Objekt zur Klasse Parameter gehört.
Parameter	Vector<Parameter>	Eingabeparameter für Submodell	Die Klasse Parameter wird in unteren Kapitel vorgestellt
Resultsparameter	Vector<Resultsparameter>	Ausgabe- und Zwischenparameter vom Submodell	Die Klasse ResultsParameter wird in unteren Kapitel vorgestellt

Tabelle 7-3    Objekte und Methoden der Klasse „Parameter“

Objekte	Typ	Beschreibung	Entsprechende Methoden
Name	String	Name des Parameters	setName(String) getName()
Unit	String	Einheit des Parameterwerts	setUnit(String) getUnit()
Description	String	Beschreibung des Parameters	setDescription(String) getDescription()
intervalMin	double	Physikalisches Werteintervall des Parameters	setIntervalMin(double) getIntervalMin()
intervalMax	Double	Physikalisches Werteintervall des Parameters	setIntervalMax(double) getIntervalMax()
distribution	String	Statistische Verteilung für Sampleerzeugung	setDistribution(String) getDistribution()
minValue	Double	Minimaler Wert bei Gleichverteilung	setMinValue(double) getMinValue()
maxValue	Double	Maximaler Wert bei Gleichverteilung	setMaxValue(double) getMaxValue()
meanValue	Double	Mittelwert bei Gaußverteilung	setMeanValue(double) getMeanValue()
Sdev	Double	Standardabweichung bei Gaußverteilung	setSdev(double) getSdev()
Value	Double	Deterministischer Wert des Parameters	setValue(double) getValue()
ifOldSample	Boolean	Entscheidet ob die Samplingswerte vom alten Rechenfall eingelesen werden soll	setIfOldSample(boolean) getIfOldSample()
samplingValues	Vector <double>	Generierte Zufallszahlen für Parameter	Die Zufallszahlen, die später zu Realisierungen eingesetzt werden

Tabelle 7-4 Attribute und Methode der Klasse „ResultsParameter“

Objekte	Typ	Beschreibung	Entsprechende Methoden
Name	String	Name des Parameters	setName(String) getName()
Unit	String	Einheit des Parameters	setUnit(String) getUnit()
Description	String	Beschreibung des Parameters	setDescription(String) getDescription()
FileName	String	Name der Datei, aus der die Werte gelesen werden	setFileName(String) getFileName()
ifDisplay	Boolean	Ob der Parameter grafisch dargestellt werden soll	setIfDisplay(bool) getIfDisplay()
ifCouple	Boolean	Ob der Parameter als Eingabeparameter fürs nächstes Modell	setIfCouple(bool) getIfCouple()
calculate-ResultsValues	2D-Vector<double>	Die zeitabhängigen Rechenergebnisse	
calculateResultsValuesForStatistic	2D-Vector<double>	Die Rechenergebnisse für statistische Datenverarbeitung	
minimalValue	Vector<double>	Berechnete Min-Werte für das Rechenergebnis	
maximalValue	Vector<double>	Berechnete Max-Werte für die Darstellung	
firstQuartile	Vector<double>	25%-Quantil für die Ergebnisreihe	
Median	Vector<double>	50%-Quantil für die Ergebnisreihe	

Objekte	Typ	Beschreibung	Entsprechende Methoden
thirdQuartile	Vector<double>	75%-Quantil für die Ergebnisreihe	
Mean	Vector<double>	Mittelwerte der Ergebnisreihe	
Sdev	Vector<double>	Standard-abweichung der Ergebnisreihe	
maxPerRealization	Vector<double>	Der maximale Wert jeder Realisierung	
Integral	Vector<double>	Integration von Werten über die Zeitreihe jeder Realisierung	

Tabelle 7-5 Attribute und Methode der Klasse „ParameterDeterministic“

Objekte	Typ	Beschreibung	Entsprechende Methoden
Name	String	Name des Parameters	setName(String) getName()
Unit	String	Einheit des Parameters	setUnit(String) getUnit()
Description	String	Beschreibung des Parameters	setDescription(String) getDescription()
minValue	Double	Minimaler Wert des Parameters	setMinValue(double) getMinValue()
maxValue	Double	Maximaler Wert des Parameters	setMaxValue(double) getMaxValue()
Value	Double	Deterministischer Wert des Parameters	setValue(double) getValue()

Tabelle 7-6 Attribute und Methode der Klasse „Geoformation“

Objekte	Typ	Beschreibung	Methoden
Name	String	Name der Geoformation des Modells	setName(String) getName()
Description	String	Beschreibung der Geoformation des Modells	setDescription(String) getDescription()
Length	Double	Die Länge (Geometrie) der geologischen Formation	setLength(double) getLength()
elementLength	Double	Die Länge der Modellelemente des FEM-Gitters	setElementLength(double) getElementLength()
Width	Double	Die Breite (geometrie) der Geoformation	setWidth(double) getWidth()
elementWidth	Double	Die Breite der Modellelemente des FEM-Gitters	setElementWidth(double) getElementWidth()
Thickness	Double	Die Dicke (Geometrie) der Geoformation	setThickness(double) getThickness()
elementThickness	Double	Die Dicke der Modellelemente des FEM-Gitters	setElementThickness(double) getElementThickness()
Permeability	Double	Permeabilität der Geoformation	setPermeability(double) getPermeability()
Porosity	Double	Porosität der Geoformation	setPorosity(double) getPorosity()
Storativity	Double	Speicherfähigkeit der Geoformation	setStorativity(double) getStorativity()

Objekte	Typ	Beschreibung	Methoden
rockDensity	Double	Dichte der Geoformation	setRockDensity(double) getRockDensity()
Dispersions-length	Double	Dispersionslänge der Geoformation	setDispersionslength(double) getDispersionslength()
Kd_Value	Double	Adsorptionskoeffizient (Kd) der Formation	setKd(double) getKd()
GeologicAnomaly	Vektor<GeologicAnomaly>	Die Anzahl der Geoanomalien in der Geoformation	setAnomalyNumber(int) getAnomalyNumber()

Tabelle 7-7 Objekte und Methoden der Klasse „GeologicAnomaly“

Objekte	Typ	Beschreibung	Entsprechende Methoden
Description	String	Kurze Beschreibung der Anomalie	setDescription(String) getDescription()
minXCoordinate	Double	x-Koordinate der „rechte“ Seite	setMinXCoordinate(double) getMinXCoordinate()
maxXCoordinate	Double	x-Koordinate der „linke“ Seite	setMaxXCoordinate(double) getMaxXCoordinate()
minYCoordinate	Double	y-Koordinate der „untere“ Seite	setMinYCoordinate(double) getMinYCoordinate()
maxYCoordinate	Double	y-Koordinate der „obere“ Seite	setMaxYCoordinate(double) getMaxYCoordinate()

Objekte	Typ	Beschreibung	Entsprechende Methoden
isAnexistingFormation	Boolean	Entscheidet ob die Geoanomalie die gleiche Eigenschaft wie eine Formation hat	setIsAnexistingFormation(bool) getIsAnexistingFormation()
existingFormationNumber	integer	Angabe dass die Geoanomalie identische Eigenschaft von einer Formation übernimmt	setExistingFormationNumber(int) getExistingFormationNumber()
Permeability	Double	Permeabilität der GeoAnomalie	setPermeability(double) getPermeability()
Porosity	Double	Porosität der GeoAnomalie	setPorosity(double) getPorosity()
Storativity	Double	Speicherfähigkeit der GeoAnomalie	setStorativity(double) getStorativity()
rockDensity	Double	Dichte des Gesteins der GeoAnomalie	setRockDensity(double) getRockDensity()
Dispersionslength	Double	Dispersionslänge der GeoAnomalie	setDispersionsLength(double) getDispersionslength()
Kd_Value	Double	Absorptionskoeffizient der GeoAnomalie	setKdValue(double) getKdValue()



Tabelle 7-8     Attribute und Methoden der Klasse „Regression“

Objekte	Typ	Beschreibung	Methoden
X_Value	Double	Berechnung der Summe von Samples der Eingabe	setX_Value() getX_Value()
Y_Value	Double	Berechnung der Summe der „Quantity of interest“ der Ausgabe	setY_Value() getY_Value()
mean_X	Double	Mittelwert der Samples der Eingabe	setMean_X() getMean_X()
mean_Y	Double	Mittelwert der „Quantity of interest“ der Ausgabe	setMean_Y() getMean_Y()
Sxx	Double	Quadratsumme der Eingabe	setSxx() getSxx()
Syy	Double	Varianz der „Quantity of interest“ der Ausgabe	setSyy() getSyy()
Sxy	Double	Produktsumme der „Quantity of interest“ der Ausgabe und Eingabesamples	setSxy() getSxy()
a_Value	Double	Steigung der Regressionsgeraden	setA_Value() getA_Value()
b_Value	Double	Schnittpunkt der Regressionsgeraden mit der y-Achse	setB_Value() getB_Value()
Bestimmtheitsmass	Double	Bestimmtheitsmaß zwischen den Ein- und Ausgabeparametern	setStorativity(double) getStorativity()

	Seite
Tabelle 4-1 Referenzwerte für die probabilistischen Parameter im Level-E Modell	48
Tabelle 4-2 Probabilistischen Parameter der Level-E Modellkomponenten	49
Tabelle 4-3 Lineare/Nichtlineare Adsorptionsisotherme [30]	66
Tabelle 4-4 Einstellung der Referenzwerte für die probabilistischen Parameter	72
Tabelle 4-5 Verteilungen mit Verteilungsmaßen für die probabilistischen Parameter	75
Tabelle 4-6 Einstellung der Referenzwerte für die probabilistischen Parameter	88
Tabelle 4-7 Deterministische Werteinstellung der Parameter	90
Tabelle 4-8 Verteilungseinstellung für die probabilistischen Parameter	92
Tabelle 4-9 Einstellung der Referenzwerte der probabilistischen Parameter	107
Tabelle 4-10 Verteilungseinstellung für die probabilistischen Parameter	109
Tabelle 5-1 Probabilistische Parameter für die Simulation eines Transportmodells	136
Tabelle 5-2 Deterministische Parameter eines Transportmodells	138
Tabelle 7-1 Objekte und Methoden der Klasse „Model“	164
Tabelle 7-2 Objekte und Methoden der Klasse „SubModel“	166
Tabelle 7-3 Objekte und Methoden der Klasse „Parameter“	167
Tabelle 7-4 Attribute und Methode der Klasse „ResultsParameter“	168
Tabelle 7-5 Attribute und Methode der Klasse „ParameterDeterministic“	169
Tabelle 7-6 Attribute und Methode der Klasse „Geoformation“	170
Tabelle 7-7 Objekte und Methoden der Klasse „GeologicAnomaly“	171
Tabelle 7-8 Attribute und Methoden der Klasse „Regression“	173

## 9                    **Abbildungsverzeichnis**

Abb. 2.1	Nutzeroberfläche von AMBER und ein einfaches Beispiel von Schadstofftransport /Bildschirmausschnitt der AMBER-GUI/	11
Abb. 2.2	Fenster zur Definition für die Parametereigenschaften im Modell /Bildschirmausschnitt der AMBER-GUI/	13
Abb. 2.3	Fenster zur Definition der mathematischen Beziehungen zwischen den Parametern /Bildschirmausschnitt der AMBER-GUI/	13
Abb. 2.4	Definitionsfenster für die Modellkomponenten /Bildschirmausschnitt der AMBER-GUI/	14
Abb. 2.5	Fenster der grafischen Darstellungsmöglichkeiten /Bildschirmausschnitt der AMBER-GUI/	15
Abb. 2.6	Nutzeroberfläche von GoldSim. Hier werden die Komponenten eines Modells erstellt /Bildschirmausschnitt der GoldSim-GUI/	16
Abb. 2.7	Ein Beispiel für ein Kluft-Transportsystem, das durch Kombination von „Pipe Pathway“-Komponenten modelliert wird [16]	18
Abb. 2.8	Multi-Ebenen-Kopplungsarchitektur von ALLIANCES [14]	21
Abb. 3.1	Arbeitsprozedur von RESUS	23
Abb. 3.2	Arbeitsprozeduren mit der GUI basierten Methode.	24
Abb. 3.3	Modellkette eines Simulationsmodells bei RESUS	26
Abb. 4.1	Anfangsphase bei Aufruf der RESUS-GUI	27
Abb. 4.2	Tabs-Aufbau der RESUS-Oberfläche	28
Abb. 4.3	Auswahl vom Modell „Ishigami“ bei dem ersten Menü-Tab	29
Abb. 4.4	Tabelle für die Festlegung der Referenzwerte der probabilistischen Parameter	30
Abb. 4.5	Warnungsfenster beim Anklicken der Tabellenzelle „Description“	30
Abb. 4.6	Dialogfenster bei der Eintragung eines Parameterwertes außerhalb des gültigen Wertebereichs	31
Abb. 4.7	RESUS-Tab für die Einstellung der entsprechend ihrer Verteilung zufallsgenerierten Parameterwerte der probabilistischen Parameter	32
Abb. 4.8a	Fenster für die Einstellung der Verteilungsfunktion eines probabilistischen Parameters bei der Wahl der Verteilungen „uniform“ und „log_uniform“	32

Abb. 4.9	Fenster zur Eingabe der Werte für die Dreieckverteilung	34
Abb. 4.10	Fenster bei Verletzung des festgelegten Wertintervalls	34
Abb. 4.11	Warnung bei Inkonsistenz der hinterlegten Werte (Minimal-, Maximal- oder, Modalwert)	35
Abb. 4.12	Fenster für die Einstellung der statistischen Verteilung eines probabilistischen Parameters	35
Abb. 4.13	Anzeige einer Warnung, wenn der festgelegte Mittelwert der Gaußverteilung außerhalb des vorgegebenen Wertintervalls liegt	36
Abb. 4.14	Tab für die Einstellung für die Darstellung der Ausgabeparameter der Simulation	37
Abb. 4.15	Tab „Statistic Display“	38
Abb. 4.16	Histogramm aus 40 Wertklassen mit der Ergebnisaufbereitungsmethode „Max per realization“	39
Abb. 4.17	Histogramm aus 50 Wertklassen mit der Ergebnisaufbereitungsmethode „Max per realization“	39
Abb. 4.18	Rechts, unten: Auswahl des Eingangsparameters zum Streudiagramm (Scatterplot). Alle probabilistischen Parameter stehen zur Auswahl	40
Abb. 4.19a	Streudiagramm zum Ishigami-Modell, $X_i$ gegen $Y$ , links ist die Darstellung in RESUS, Rechts ist die Darstellung der Ergebnisse aus der Literatur [22]	41
Abb. 4.20	Aufbau eines Level-E Modells eines stark vereinfachten Endlagersystems[24]	44
Abb. 4.21	Auswahl des Level-E Modells im „Model Selection“-Tab	47
Abb. 4.22	Tabelle zur Einstellung der deterministischen Werte der probabilistischen Parameter	47
Abb. 4.23	Tabelle zur Darstellung der 30 Zufallswerte des Level-E-Modells	50
Abb. 4.24	Tabelle für das „Output Control“	51
Abb. 4.25	Bildschirm-Ausschnitt beim Aufruf vom Level-E Solver: „gtmle.exe“	51
Abb. 4.26	Tab „Graphic Plot“ auf der RESUS-GUI zur grafischen Darstellung der zeitabhängigen Ergebnisdaten	52
Abb. 4.27	Darstellung von 301 Gesamtdosis-Zeitverläufen, berechnet aus 301 verschiedenen Realisierungen	53
Abb. 4.28a	Zeitlicher Verlauf der Konzentration von I-129 am Endpunkt der zweiten Geoformation.	54

Abb. 4.29	Darstellung der statistischen Maße als dicke Linien über den Realisierungen	55
Abb. 4.30	Histogramm der Schadstoffgesamtdosis, 30 Wertklassen	56
Abb. 4.31	Histogramm des I-129 Konzentrationsflusses am Ende der zweiten Geoschicht, 30 Wertklassen	57
Abb. 4.32	Histogramm des Np-237 Konzentrationsflusses am Endpunkt der zweiten Geoformation, 30 Wertklassen	58
Abb. 4.33	Anzeige einer Warnung, wenn bei der Histogrammanalyse eine logarithmische Transformation mit Nullen oder negativen Werten durchgeführt wird	59
Abb. 4.34	Streudiagramm: logarithmierte Wasserfließrate in der ersten Geoschicht gegen die Gesamtdosis in der Biosphäre	59
Abb. 4.35	Streudiagramm: logarithmierte Wasserfließrate in der ersten Formation gegen den logarithmierten Konzentrationsfluss von I-129 am Endpunkt der ersten Geoschicht	60
Abb. 4.36	Streudiagramm: logarithmierte Wasserfließrate in der ersten Geoschicht gegen den logarithmierten Konzentrationsfluss von U-233 am Endpunkt der ersten Geoschicht	61
Abb. 4.37	Streudiagramm: logarithmierte Wassermenge in der Biosphäre gegen die logarithmierte Gesamtdosis	61
Abb. 4.38	Streudiagramm: logarithmierte Wasserfließrate in der zweiten Geoschicht gegen die logarithmierte Gesamtdosis	62
Abb. 4.39	Streudiagramm: logarithmierte Wasserfließrate in der zweiten Geoformation gegen die logarithmierte Ausflussrate von U-233 in der ersten Geoformation	63
Abb. 4.40	Schematische grafische Darstellung eines Deckgebirges auf einem Salzstock in Deutschland [40]	64
Abb. 4.41	Eindimensionales Transportmodell eines Deckgebirges als ein Beispiel	65
Abb. 4.42	Grafische Darstellung des eindimensionalen Transportmodells in RESUS	68
Abb. 4.43	Deckgebirge des Bergwerks Asse mit zwei möglichen Transportpfaden (Pfad I und II) vom Grubengebäude bis zur Erdoberfläche [42]	69
Abb. 4.44	Das erste Tab der Benutzeroberfläche für die Modellauswahl eines eindimensionalen Transportmodells	71
Abb. 4.45	Option „Deterministic“ als Verteilung und das „Use the old samples“ Häkchen	74
Abb. 4.46	Tab für die Ausgabekontroll	77

Abb. 4.47	Konzentration von Pu-239 am Ende des Transportpfads	78
Abb. 4.48	Konzentration von U-235 am Ende des Transportpfads. Die rote Kurve markiert die maximalen Werte aller Realisierungen	78
Abb. 4.49	Konzentration von Pa-231 am Ende des Transportpfads. Die rote Kurve markiert die maximalen Werte aller Realisierungen	79
Abb. 4.50	Histogramm der maximalen Werte der Konzentration von U-235 mit 20 Wertklassen. Die Quantile sind im Textfeld rechts oben abzulesen	80
Abb. 4.51	Histogramm der maximalen Werte der Konzentration von Pa-231 mit 20 Wertklassen	80
Abb. 4.52	Streudiagramm: Probabilistische Porositätswerte des Transportpfads gegen die maximalen Konzentrationen aller Rechenrealisierungen von U-235	81
Abb. 4.53	Streudiagramm (Scatterplot): probabilistische Porositätswerte gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfads	82
Abb. 4.54	Streudiagramm (Scatterplot): probabilistische $K_d$ -Werte gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfads	83
Abb. 4.55	Streudiagramm (Scatterplot): probabilistische Weglängen des Transportpfades gegen die maximalen Konzentrationen von U-235 am Ende des Transportpfads	83
Abb. 4.56	Streudiagramm (Scatterplot): probabilistische Weglängen des Transportpfades gegen die maximalen Konzentrationen von Pa-231 am Ende des Transportpfads	84
Abb. 4.57	Beispiel eines zweidimensionalen Transportmodells mit zwei geologischen Schichten und zwei Geoanomalien	85
Abb. 4.58	Zweidimensionales Beispielmmodell zur Demonstration in RESUS	86
Abb. 4.59	Modellauswahl- und Einstellung im ersten Tab für das zweidimensionale Transportmodell	87
Abb.4.60	„Parameter Reference Case“-Tab zur Angabe der Referenzwerte der probabilistischen Parameter	88
Abb. 4.61	Tab „Parameter Deterministic“ für die Werteinstellung der deterministischen Parameter	90
Abb. 4.62	Tab „Parameter_Probabilistic“ für die Einstellung der Verteilungen der Parameter	92
Abb. 4.63	„Output_Control“ für den einzigen Ausgabeparameter dieses Beispiels	93
Abb. 4.64	Konzentrationsverläufe am Messpunkt von 100 Rechenrealisierungen	94

Abb. 4.65	Histogramm mit 10 Wertklassen der Konzentrationshäufigkeiten	95
Abb. 4.66a	Streudiagramm: Porosität der Geoformation [-] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m <sup>3</sup> ]	96
Abb. 4.67a	Streudiagramm: Logarithmus der Permeabilität der Geoformation [log10 m <sup>2</sup> ] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m <sup>3</sup> ]	97
Abb. 4.68	Streudiagramm: K <sub>d</sub> -Werte der Geoformation[-] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m <sup>3</sup> ]	99
Abb. 4.69	Streudiagramm: Diffusionskoeffizient [m <sup>2</sup> /s] gegen die Nuklidkonzentration von U-235 am Messpunkt [kg/m <sup>3</sup> ]	100
Abb. 4.70	Grafische Darstellung des Modellaufbaus mit drei Submodellen	102
Abb. 4.71	Beschreibungsdatei unseres Modellbeispiels, benannt „RESUS_Example“	105
Abb. 4.72	Modellauswahl beim ersten Tab für das dateibasierte Modellerstellungsverfahren	106
Abb. 4.73	Beispiel der Template-Eingabedatei, in der ein probabilistischer Parameter und ein Ausgabeparameter gekennzeichnet sind	106
Abb. 4.74	Tabelle im Tab für die Einstellung der Referenzwerte für die probabilistischen Parameter	107
Abb. 4.75	„Parameter_Probabilistic“ Tab auf der RESUS-GUI für die Korrekturen der statistischen Verteilungen der Parameter	108
Abb. 4.76	Tab „Output_Control“ zur Festlegung, welche Ausgabeparameter in diesem Beispiel weiter verarbeitet werden	110
Abb. 4.77	Dosiswerte des Beispiels und deren statistischen Kennwerte aus 300 Realisierungen	111
Abb. 4.78	Histogramm der maximalen Werte jeder Realisierung mit 30 Wertklassen	111
Abb. 4.79	Logarithmisches Streudiagramm: Dosiswerte gegen die Permeabilität der Kluft in Submodell 1	112
Abb. 4.80	Logarithmisches Streudiagramm: Dosis-Werte gegen die Permeabilität der Kluft in Submodell 2	113
Abb. 4.81	Streudiagramm: maximale Konzentrationen am Kluftende gegen die Porosität der Matrix in Submodell 1	114
Abb. 4.82	Kommandostelle für das Speichern des auf RESUS laufenden Modells. Der rote Kreis markiert den Kommando-Knopf für das Speichern	115
Abb. 4.83	Dialogfenster für die Dateiauswahl für die Speicherung des Modells	116
Abb. 4.84	Warnungsfenster beim Speicherversuch ohne erstelltes Modell	116

Abb. 4.85	Menüeintrag für die Wiederöffnung eines Modells	117
Abb. 4.86	Dialogfenster Zum Öffnen, Wiederherstellen, Einlesen eines Modells	118
Abb. 5.1	Vereinfachte Klassenstruktur eines Modells in RESUS	119
Abb. 5.2	Wichtigste Attribute der Klasse „Model“, in der oberen Reihe die Grundattribute, in der unteren Reihe die Zusatzattribute	120
Abb. 5.3	Attribute der Klasse „SubModel“	121
Abb. 5.4	Modellkette eines Simulationsmodells bei RESUS	122
Abb. 5.5	Attribute der Klasse „Parameter“	123
Abb. 5.6	Attribute der Klasse „ResultsParameter“	124
Abb. 5.7	Attribute der Klasse „Geoformation“	127
Abb. 5.8	Transport in zwei geologischen Formationen	128
Abb. 5.9	Zwei geologische Formationen mit unterschiedlichen Transporteigenschaften. Die rote Schicht hat zwei Anomalien: die dunkelblaue Zone und die hellblaue Zone	128
Abb. 5.10	Objekte und Methoden der Klasse „GeologicAnomaly“	129
Abb. 5.11	Zwei geologische Formationen mit unterschiedlichen Transporteigenschaften. Die rote Schicht hat zwei Anomalien: die dunkelblaue Zone und hellblaue Zone	130
Abb. 5.12	Arbeitsprozedur von RESUS: links ist die Kommando von Nutzer auf der GUI, links ist die entsprechenden Prozesse, die RESUS Quellcode ausführt	131
Abb. 5.13	Dateiformat der Eingabedatei „input.sam“ vom Level-E Modell	134
Abb. 5.14	Ausgabedatei „Geosph1.dat“ von Level-E	135
Abb. 5.15a	Nummerierungsregel des eindimensionalen Modells in RockFlow und OGS	140
Abb. 5.16	Ausschnitt der „rfd“-Datei	140
Abb. 5.17	Vorgehensweise der Simulationsphase eines Transportmodells in RESUS	141
Abb. 5.18	Ausschnitt der Datei „.PLT“	142
Abb. 5.19	Beschreibungsdatei eines Dateibasierten Modells	144
Abb. 5.20	Ausschnitt aus einem Beispiel einer Template-Eingabedatei	146
Abb. 5.21	Darstellung des Vorgehens bei der Herstellung eines dateibasierten Modells	147



Abb. 5.22	Eingabedatei für RockFlow mit zwei Randbedingungen: oben „BOUNDARY_CONDITIONS_CONCENTRATION“ aus der Template-Eingabedatei und unten „CURVES“ aus dem vorherigen Submodell	148
Abb. 5.23	Ein Beispiel von Maximalem Wert einer Realisierung	151
Abb. 5.24	Darstellung eines Ergebnisparameters mit 30 Realisierungen aus Level-E	152
Abb. 5.25	Beispiel eines Streudiagramms in RESUS. Das Beispiel entstammt der Level-E-Modellrechnung	154
Abb. 5.26	Textbeschreibung der Objekte und Attribute des „model“-Objekts in der zugehörigen Textdatei	156
Abb. 5.27	Textbeschreibung der Objekte und Attribute des Objekts „Submodel“ in der zugehörigen Textdatei	156
Abb. 5.28	Ausschnitt aus der Speicher-Textdatei des Model-Objekts	157
Abb. 5.29	Ausschnitt aus der Speicher-Textdatei des Model-Objekts	157
Abb. 5.30	Ausschnitt aus der „_Samplings“-Datei, die die gespeicherten Inputsamples in einem Datenblock enthält	158
Abb. 5.31	Beispielausschnitt einer „_Results“-Textdatei	158
Abb. 5.32	Beispielausschnitt einer „_SamResults“-Textdatei	159

- [1] Bundesrepublik Deutschland: Gesetz über die friedliche Verwendung der Kernenergie und den Schutz gegen ihre Gefahr (Atomgesetz). AtG. Ausfertigungsdatum: 23.12.1959. Vollzitat: „Atomgesetz in der Fassung der Bekanntmachung vom 15. Juli 1985 (BGBl. IS. 1565), das durch Artikel 1 des Gesetzes vom 31. Juli 2011 geändert worden ist. <http://www.gesetze-im-internet.de/bundesrecht/atg/gesamt.pdf>
- [2] Bundesrepublik Deutschland: Verordnung über den Schutz vor Schäden durch ionisierende Strahlen (Strahlenschutzverordnung - StrlSchV), Ausfertigungsdatum: 20.07.2001, zuletzt geändert durch Art. 5Vv. 11.12.2014 /2010. [http://www.gesetze-im-internet.de/bundesrecht/strlschv\\_2001/gesamt.pdf](http://www.gesetze-im-internet.de/bundesrecht/strlschv_2001/gesamt.pdf)
- [3] Bundesamt für Strahlenschutz (BfS): Abfallprognosen. <http://www.bfs.de/de/endlager/abfaelle/prognose.html>. Stand: 11.12.2014
- [4] Committee on Radioactive Waste Management: Managing Our Radioactive Waste Safely- CoRWM Recommendation to Government. CoRWM Doc 700, <http://corwm.decc.gov.uk/media/viewfile.ashx?filetype=4&filepath=/corwm/post-nov%2007%20doc%20store/documents/reports%20to%20government/nov%20and%20dec%202007/700%20-%20corwm%20july%202006%20recommendations%20to%20government.pdf>
- [5] Auswahlverfahren für Endlagerstandorte: Empfehlungen des AkEnd – Arbeitskreis Auswahlverfahren Endlagerstandorte. Dezember 2002
- [6] Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit (BMU): Programm für eine verantwortungsvolle und sichere Entsorgung bestrahlter Brennelemente und radioaktiver Abfälle, Entwurf vom 06.01.2015. [http://www.bmub.bund.de/fileadmin/Daten\\_BMU/Download\\_PDF/Nukleare\\_Sicherheit/nationales\\_entsorgungsprogramm\\_bf.pdf](http://www.bmub.bund.de/fileadmin/Daten_BMU/Download_PDF/Nukleare_Sicherheit/nationales_entsorgungsprogramm_bf.pdf)
- [7] Nuclear Energy Agency. Radioactive Waste Management Committee. Moving forward with geological disposal of radioactive waste, OECD 2008, NEA No.6433
- [8] The COUNCIL of the European Union. A community framework for the responsible and safe management of spent fuel and radioactive waste. Official Journal of the European Union, L 199/48-56
- [9] K.-J.Röhlig, H. Geckeis, K. Mengel: Endlagerung radioaktiver Abfälle, Fakten und Konzepte. Chemie in unserer Zeit, Volume 46, Issue 3. Online ISSN: 1521-3781.
- [10] K.-J.Röhlig, K. Mengel, H. Geckeis: Endlagerung radioaktiver Abfälle. Die Wirtsgesteine: Tonstein, Granit, Steinsalz. Chemie in unserer Zeit, Volume 46, Issue 3. Online ISSN: 1521-3781.
- [11] H. Geckeis, K. Mengel, K.-J.Röhlig: Endlagerung radioaktiver Abfälle, Chemie im Endlagersystem. Chemie in unserer Zeit, Volume 46, Issue 3. Online ISSN: 1521-3781.
- [12] BMU, Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit. Sicherheitsanforderungen an die Endlagerung wärmeentwickelnder radioaktiver Abfälle, Stand 30. September 2010, [http://www.bmub.bund.de/fileadmin/bmu-import/files/pdfs/allgemein/application/pdf/sicherheitsanforderungen\\_endlagerung\\_bf.pdf](http://www.bmub.bund.de/fileadmin/bmu-import/files/pdfs/allgemein/application/pdf/sicherheitsanforderungen_endlagerung_bf.pdf)
- [13] K. Röhlig, A. Luik, J. Schneider, L. Griffault, P. Gierszewski, U. Noseck, M. Navarro. Methods for safety assessment of geological disposal facilities for radioactive waste.

- OECD 2012, NEA. No. 6923 <http://www.oecd-nea.org/rwm/reports/2012/nea6923-MESA-initiative.pdf>
- [14] P. Montarnal, A. Dimier, E. Deville, E. Adam, J. Gaombalet, A. Bengaouer, L. Loth, C. Chavant: Coupling Methodology Within The Software Platform ALLIANCES, CIMNE, Barcelona, 2005
  - [15] Quintessa Limited: AMBER 5.4 Reference Guide. February 2011
  - [16] GoldSim Technology Group LLC: User's Guide: GoldSim Contaminant Transport Module
  - [17] Qt Designer Manual: <http://qt-project.org/doc/qt-5/designer-to-know.html>, Stand: 11.02.2015
  - [18] Martin von Löwis, Nils Fischbeck: Python2 – Einführung und Referenz der objektorientierten Skriptsprache, Addison-Wesley, ISBN 3-8273-1691-X
  - [19] B. Boese, R.P. Hirsekorn, R. Storck: Vergleich der Rechenprogramme LOPOS und MARNIE, Oktober 2000, GRS-A-3264
  - [20] D. Buhmann: Das Programmpaket EMOS, Ein Instrumentarium zur Analyse der Langzeitsicherheit von Endlagern. Juli 2000, GRS-159
  - [21] R. Mock: Methoden zur Datenhandhabung in Zuverlässigkeitsanalysen. Vdf Hochschulverlag AG an der ETH Zürich, ISBN 3728121827
  - [22] E. Plischke: Computing Global Sensitivity Measures from Given Samples. <http://www.immr.tu-clausthal.de/~epl/work/GlobalSA/GlobalSensitivityIndexEstimators.html>, Stand: 11.02.2015
  - [23] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola: Global Sensitivity Analysis. John Wiley & Sons Ltd, ISBN: 978-0-470-05997-5
  - [24] Nuclear Energy Agency. Psacoin level E intercomparison. OECD 1989.
  - [25] U. Noseck, V. Brendler, J. Flügge: Realistic Integration of Sorption Processes in Transport Codes for Long-Term Safety Assessments. 2012
  - [26] O. Kolditz, S. Bauer, L. Bilke, N. Böttcher, J. O. Delfs, T. Fischer, U. J. Görke, T. Kalbacher, G. Kosakowski, C. I. McDermott, C. H. Park, F. Radu, K. Rink, H. Shao, H. B. Shao, F. Sun, Y. Y. Sun, A. K. Singh, J. Taron, M. Walther, W. Wang, N. Watanabe, Y. Wu, M. Xie, W. Xu, B. Zehner (2012): *OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media*, *Environ. Earth Sci.*
  - [27] AF- Colenco AG, GRS, IfG: Schachtanlage Asse II Beschreibung und Bewertung der Stilllegungsoption Vollverfüllung, Stand 01. Oktober 2009.
  - [28] K. Rink, OpenGeoSys data Explorer Manual
  - [29] Shao, H.B., Dmytrieva, S.V., Kolditz, O., Kulik, D.A., Pfingsten, W., Kosakowski, G., (2009): Modeling reactive transport in non-ideal aqueous-solid solution system *Appl. Geochem.* 24 (7), 1287 – 130.
  - [30] Habbar, A.: Direkte und inverse Modellierung reaktiver Transportprozesse in klüftig-porösen Medien, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, 31.10.2001
  - [31] Simunek, J., Sejna, M. Van Genuchten, M. Th.: TheHYDRUS 2D software package for simulating the two-dimensional movement of water, heat, and multiple solutes in

- variably saturated media, version 2.0, U.S. Salinity Laboratory, Agricultural Research Service, U.S. Department of Agriculture, Riverside, California, 1999.
- [32] Kolditz, O., Habbar, A., Kaiser, R., Kasper, H., Rother, T., Schulze-Ruhfus, M., Thorenz, C. & Zielke, W.: Software Concept of Simulating Coupled Processes in Subsurface Hydrosystems. In: Proceedings of the 3rd International Conference on Hydroinformatics, pp 613-618, Copenhagen, Denmark, August 24-26, 1998. Balkema, Rotterdam/Brookfield, ISBN 90 5410 983 1.
  - [33] Kolditz, O., Habbar, A., Kaiser, R., Schulze-Ruhfus, M. & Thorenz, C.: ROCKFLOW - Theory and Users Manual. Release 3.2, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, 1998.
  - [34] M. Kohlmeier, J. Maßmann, M. Wulkau, G. Ziefle: RockFlow 5 User's Manual, 12th January 2009
  - [35] R. Kaiser, A. Habbar, Olaf Kolditz: RockFlow Tutorial. Version 0.1.5 für RockFlow 3.8.35+
  - [36] Bundesamt für Strahlenschutz: Vom Salzbergwerk zum Atomlager. 10.10.2012. [http://www.asse.bund.de/DE/2\\_WasIst/Geschichte/\\_node.html](http://www.asse.bund.de/DE/2_WasIst/Geschichte/_node.html)
  - [37] G. Bracke, L. Armean, W. Müller / ISTec(2003): Ermittlung der Gasbildung in der Schachtanlage Asse. Kammer- und bereichsspezifische Berechnung bis 150 000 Jahre sowie vollständiger Umsetzung (2003)
  - [38] G. Bracke, W. Müller / ISTec(2005): Realistische und maximale Gasbildung in der Schachtanlage Asse (2005)
  - [39] P. Roussev, O. Dyogtyev, R. Wolters, K.-H. Lux, Risikobetrachtung für eine längerfristige Offenhaltung der Asse, Teil Gebirgsmechanische Untersuchung
  - [40] A. Poller, G. Resele, J. Poppei, Deckgebirgsmodellierung Phase IV Grundwasserbewegung im Deckgebirge des Standortes Asse, Schlussbericht, September 2006
  - [41] W. Klemenz, J.-M. Lavanchy, G. Resele, A. Poller, Hydrogeologische Modellvorstellungen, Bericht 4956/07, Revision 3, November 2006
  - [42] C. Küppers, V. Ustohalova, M. Steinhoff: Neuberechnung zu den Auswirkungen eines auslegungsüberschreitenden Lösungszutritts in der Schachtanlage Asse II – Weiterentwicklung der radioökologischen Modellierung. 16.06.2010
  - [43] J. Poppei, J. Holocher, J. Brommundt: Schachtanlage Asse II- Abschätzung der Dosis bei einem unterstellten Absaufen des Grundgebäudes, 10. Oktober 2008
  - [44] J. Poppei, J. Holocher, J. Brommundt: Schachtanlage Asse II - Abschätzung der Trinkwasserdosis bei einem unterstellten Absaufen des Grubengebäudes, 8. Mai 2009
  - [45] D. Buhmann, B. Förster, J. Mönig: Abschätzung potenzieller Strahlenexpositionen in der Umgebung der Schachtanlage Asse II infolge auslegungsüberschreitender Zutrittsraten der Deckgebirgslösung während der Betriebsphase, 21. April 2009
  - [46] J. Mönig, Schachtanlage Asse: Stellungnahme zur Wirksamkeit von Einzelmaßnahmen der Notfallplanung
  - [47] L. Moreno, I. Neretneiks, Flow and nuclide transport in fractured media: the importance of the flow-wetted surface of radionuclide migration, Journal of Contaminant Hydrology. Volume 13, Issues 1-4, June 1993, Pages 49-71.

- [48] D. C. -F. Shih, Uncertainty analysis: one-dimensional radioactive nuclide transport in a single fractured media, Stochastic Environmental Research and Risk Assessment (2004) 18: 198-204
- [49] Y. Wu, M. Ye, E. A. Sudicky, Fractured-Flow-Enhanced matrix diffusion in solute transport through fractured porous media, transport in porous media, January 2010, Volume 81, Issue 1, pp 21-34
- [50] Bjarne Stroustrup: The C++ Programming Language, Addison-Wesley, 4<sup>th</sup> Edition.
- [51] Qt Creator Manual: <http://qt-project.org/doc/qtcreator-2.8/> 11.02.2015
- [52] StochasticLib C++ Bibliothek: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>
- [53] G. Moridis, Y. Wu, K. Pruess, EOS9nT: A TOUGH2 module for the simulation of water flow and solute/colloid transport in the subsurface, Earth Sciences Division, Lawrence Berkeley National Laboratory, Berkeley March 1999
- [54] David L. Parkhurst, C.A.J. Appelo: Description of Input and Examples for PHREEQC
- [55] H. Witlox: Overview of consequence modelling in the hazard assessment package Phast, DNV Software, London UK.
- [56] K. Röhlig, E. Plischke, X. Li: Probabilistische Methoden als Hilfsmittel zur Bemessung von Verschlussbauwerken für ein Endlager im Salinar. Institut für Endlagerforschung, TU-Clausthal, Januar 2014
- [57] M. Schild: Verbundene Mikroporosität in Kristallgesteinen Fallstudie Felslabor Grimsel. Dissertation, Mathematisch-Naturwissenschaftlichen Fakultäten der Georg-August-Universität zu Göttingen, 04-11-1999
- [58] IAEA: Natural activity concentrations and fluxes as indicators for the safety assessment of radioactive waste disposal. October 2005
- [59] Website: <http://www.agner.org/random/>, Stand: 11.02.2015
- [60] T. Bedford: Use and abuse of expert judgement in risk studies. Department of Management Science, University of Strathclyde.
- [61] Qwt User' Guide 6.1.2, Website: <http://qwt.sourceforge.net/>, Stand: 11.02.2015
- [62] Bundes Republik Deutschland: Gesetz zur Suche und Auswahl eines Standortes für ein Endlager für Wärme entwickelnde radioaktive Abfälle (Standortauswahlgesetz-StandAG), Ausfertigungsdatum: 23.07.2013: <http://www.gesetze-im-internet.de/bundesrecht/standag/gesamt.pdf>, Stand: 11.02.2015
- [63] BMU (Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit): Programm für eine verantwortungsvolle und sichere Entsorgung bestrahlter Brennelemente und radioaktiver Abfälle (Nationales Entsorgungsprogramm), Entwurf vom 06.01.2015:  
[http://www.bmub.bund.de/fileadmin/Daten\\_BMU/Download\\_PDF/Nukleare\\_Sicherheit/nationales\\_entsorgungsprogramm\\_bf.pdf](http://www.bmub.bund.de/fileadmin/Daten_BMU/Download_PDF/Nukleare_Sicherheit/nationales_entsorgungsprogramm_bf.pdf)